Petri Nets for Systems Concurrent Engineering

Giuliani Paulineli Garbi[a,1] , Geilson Loureiro[b]

[a]Engineering Department, College Anhanguera of São José (São José dos Campos), BRA.
[b]Brazilian Institute for Space Research (INPE) and Technological Institute of Aeronautics (ITA).

Abstract. The paper proposes a generic model to represent the workflow based on Petri Nets theory of the activities used in the Systems Concurrent Engineering process and to use Petri Nets to support decision making when architecting a complex system. Systems Concurrent Engineering is a method that integrates systems engineering and concurrent engineering in the same integrated product development framework. The framework is applicable to the development of complex products. The properties of the Petri Nets allow the design of the generic model through formal language, semantic formalism and techniques to supporting analysis of process and architecture performance through graphical and algebric tools.

Currently to maintain the competitiveness the main productive segments must seek to apply methods to innovate and develop their complex products and services with lower costs, improved productivity and quality, and in less time. Addressing the needs of productive segments, the paper presents a generic model to support and encourage the development of complex products and services. The main benefit of the generic model is that despite the actual various ways of implementing a system and of performing the systems concurrent engineering process, Petri Nets would allow to assess the process and architecture alternatives at very early stages of a complex product development process, based only on the process and product Petri Net models.

Keywords. Systems Concurrent Engineering, Petri Nets, Systems Engineering Process, Concurrent Engineering, System Architecture

---

[1] Engineering Department, College Anhanguera of São José, avenue Dr. João Batista de Sousa Soares, 4121, Jardim Morumbi, São José dos Campos, São Paulo, Brazil, CEP: 12223-660; +55 (12) 3512-1300; Fax: +55 (12) 3512-1316; Email: giuliani.garbi@unianhanguera.edu.br; http://www.unianhanguera.edu.br

D. D. Frey et al. (eds.), *Improving Complex Systems Today*,
Advanced Concurrent Engineering, DOI: 10.1007/ 978-0-85729-799-0_8,

1 Introduction

This paper concerns a generic model to support decision making in architectural reasoning for complex socio-technical systems. The architecture term denotes the stable properties of the system of interest. The architectural reasoning is defined as a transformative process that utilizes knowledge about stable properties in a system to achieve certain global objectives. The complex socio-technical systems refers to systems involving multiple stakeholders and requiring multiple knowledge domains [6].

In the process of architecting complex socio-technical systems that involves multiple stakeholders and knowledge domains, to assess the architecture alternatives at very early stages of a development process often becomes a considerable challenge. This challenge presents two interrelated opportunities. First, a domain-independent architectural reasoning techniques that can be implemented computationally over multiple disciplines and second, identifying a single formal language and the techniques analysis tools to support Systems Concurrent Engineering process.

Therefore this paper proposes a generic architecture that represents a workflow based on Petri Nets theory to Systems Concurrent Engineering process. The main purpose of workflow is to support the definition, execution, registration and control processes, and the development with Petri Nets allows the construction of a single formal language and the techniques analysis tools to support analysis of process performance because it is a combination of specification of oriented events and states with excellent graphics tools [3, 5].

The paper presents in Section 2 the Systems Concurrent Engineering approach that integrated systems engineering and concurrent engineering process for integrated complex product development. Section 3 presents the main concepts of Petri Nets. Section 4 presents the generic architecture that represent a workflow based on Petri Nets theory to the Systems Concurrent Engineering process and Section 5 draws some conclusions.

2 Systems Concurrent Engineering

The Systems Concurrent Engineering is a modeling framework that integrates the product and their performing organizations [1, 2]. Stakeholder analysis, requirements analysis, functional analysis and implementation or physical analysis processes are carried out through the simultaneous modeling of product and organization, at all levels of the product hierarchy, deriving attributes as emergent properties of a whole integrated system [7, 8, 9].

Figure 1 presents the total view framework, it has three dimensions. Figure 2 provides an overview of the 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. Figure 3 details the concurrent structured analysis method showing how to incorporate the concurrent engineering concept in the systems engineering process.

Step 1: identify the product mission, the product life cycle processes and their scenarios and, the scope of the development effort. The scope of the development effort consists of the life cycle processes or their scenarios that the development organization is also responsible for accomplishing.

Step 2: identify product stakeholders and their concerns for each product life cycle process scenario. Identify organization stakeholders and their concerns for each process 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. Requirement analysis transforms stakeholder requirements into system requirements.

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. 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.

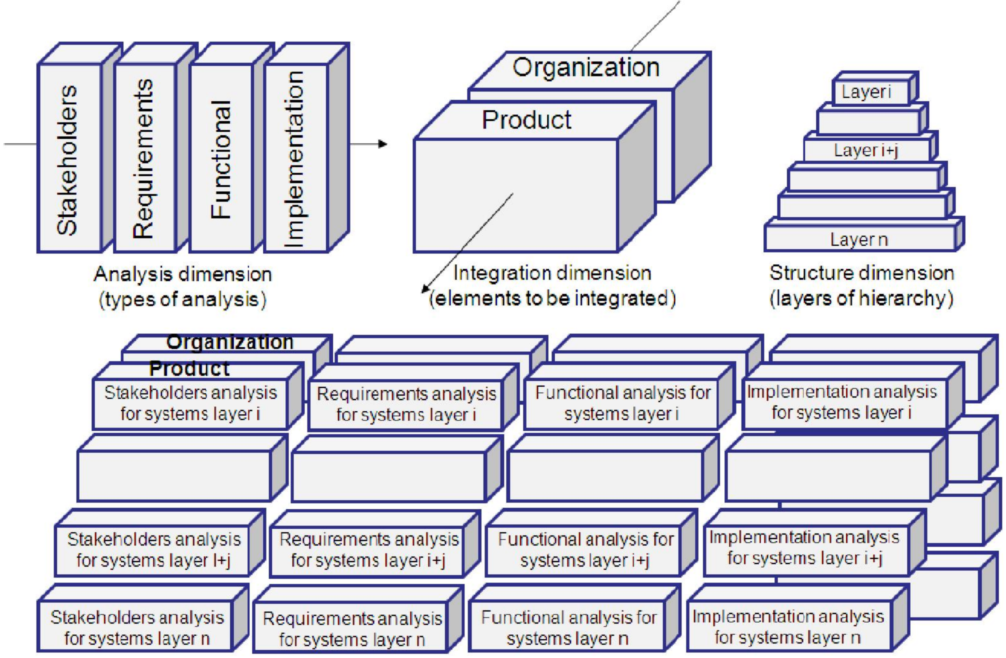


Figure 1. A framework to address complexity in complex product development – the total view framework. Source: [7], [8]

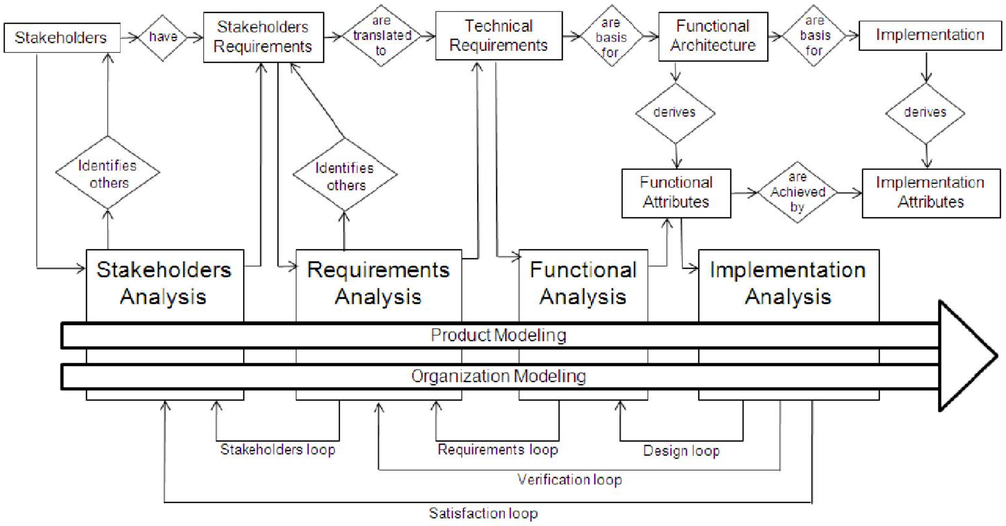


Figure 2. A method within the total view framework – the concurrent structured analysis method. Source: [7], [8]

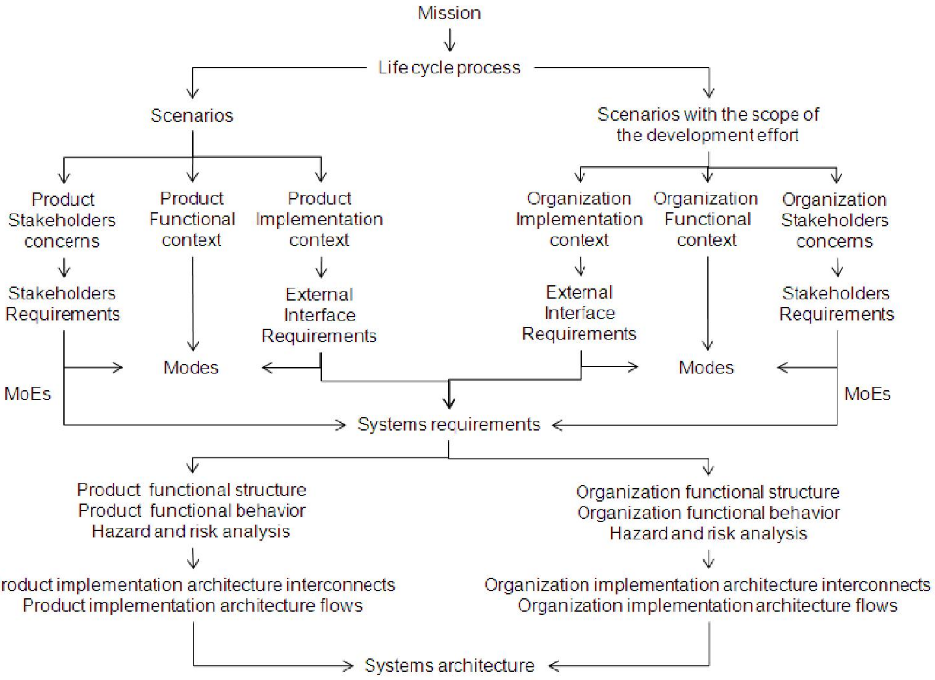


Figure 3. The system concurrent engineering method in detail. Source: [7], [8]

3 Petri Nets

The concept of Petri Nets was introduced by Carl Adam Petri in his doctoral thesis in 1962. It is a modeling technique that allows the representation of systems through its graphical and algebric formalism. The technique has properties that allow to model parallel systems, concurrent, asynchronous and non-deterministic, and has mechanisms that treat the hierarchy design and high level of abstraction that are fundamental to the development of complex systems. During the past 20 years, Petri Nets have been applied in many applications in different areas, currently there are many commercial and academic tools for design, simulation, and analysis system based on Petri Nets [4, 5].

Petri Net is a model of the state-event type, where each event possesses daily pre-conditions to allow its occurrence and pos-conditions of this event, illustrated in Figure 4. It is also seen as a particular type of guided graph that allows modeling the static properties of a system to the discrete events: transitions (events that characterize the changes of state in the system), and the places (conditions against which the events must be certified in order to happen) linked by directed weighed arcs. The transition is triggered only if there is at least one marking or fiche (token) in place proceeding of transition.

Petri Net is, therefore, a formalism that allows the modeling of discrete dynamic systems with great power of expressiveness, allowing to represent with easiness all the relations of causalities between the processes in situations of sequence, conflict, parallelism and synchronization. Figure 4 provides an overview of Petri Nets graphical tools.

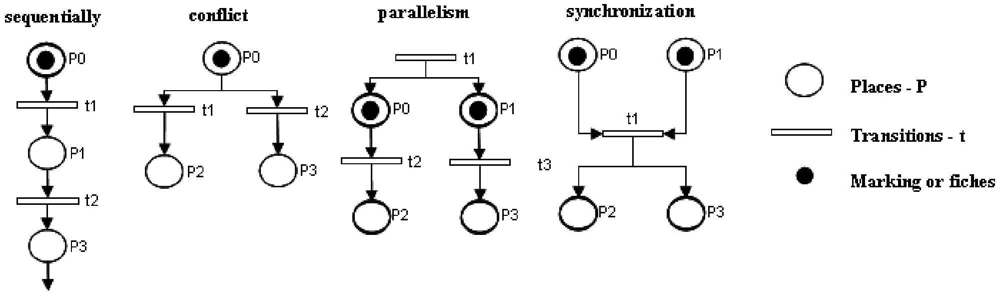


Figure 4. Petri Nets graphical tools. Source: [3]

A Petri Net (simple or autonomous) is composed of five parts: a set of places *P*, a set of transitions *t*, an application of input *I*, an application of exit *O* and a set of markings *M* that represent the markings of places *P*, illustrated in the Equation 1.

$$R = (P, T, I, O, M) \tag{1}$$

4 Petri Nets for Systems Concurrent Engineering

Figure 5 presents the Petri Net graph for the Systems Concurrent Engineering process. The Figure 5 represents the generic model of the concurrent structured analysis method workflow using Petri Nets notation. The stages of the system architecting process illustrated in Figure 3 are defined by the workflow of the places and the transitions.

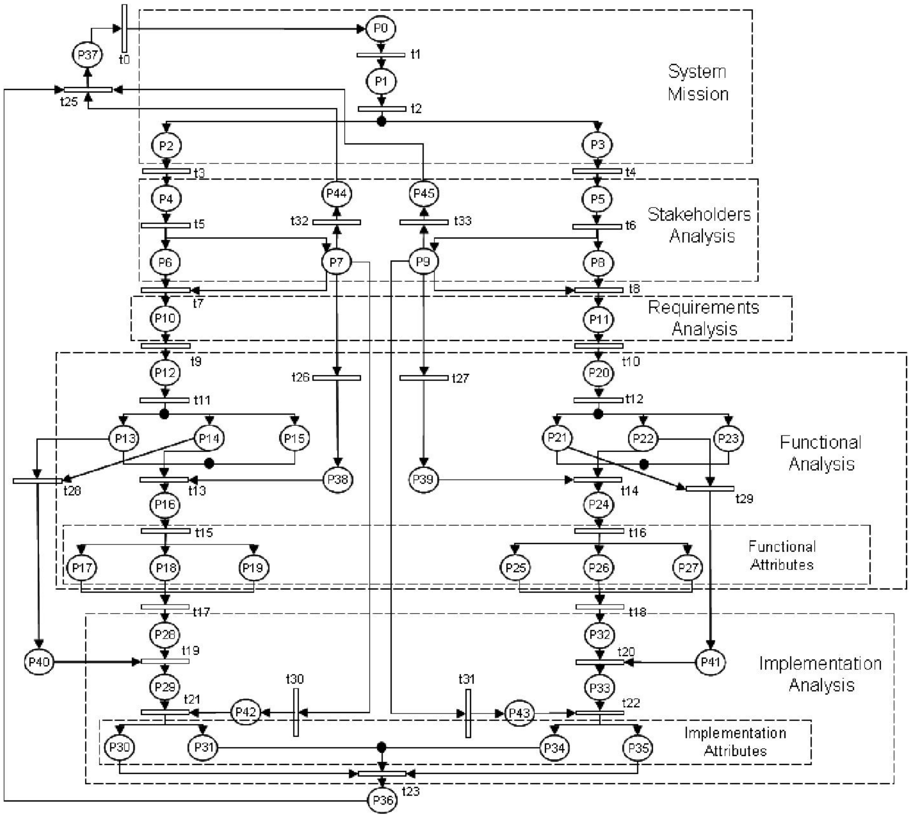
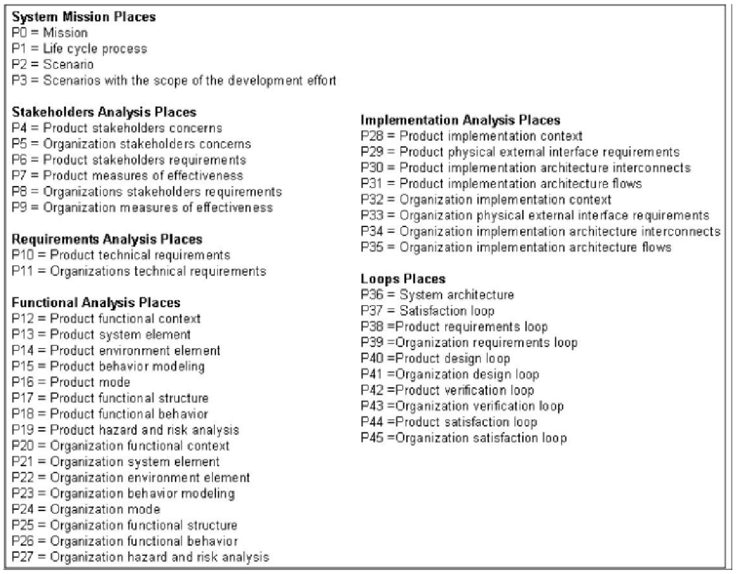


Figure 5. Petri Nets graph for Systems Concurrent Engineering process



System Mission Places
P0 = Mission
P1 = Life cycle process
P2 = Scenario
P3 = Scenarios with the scope of the development effort

Stakeholders Analysis Places
P4 = Product stakeholders concerns
P5 = Organization stakeholders concerns
P6 = Product stakeholders requirements
P7 = Product measures of effectiveness
P8 = Organizations stakeholders requirements
P9 = Organization measures of effectiveness

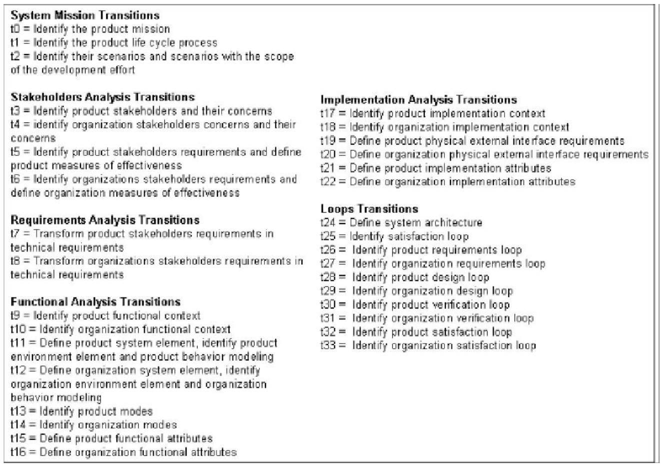
Requirements Analysis Places
P10 = Product technical requirements
P11 = Organizations technical requirements

Functional Analysis Places
P12 = Product functional context
P13 = Product system element
P14 = Product environment element
P15 = Product behavior modeling
P16 = Product mode
P17 = Product functional structure
P18 = Product functional behavior
P19 = Product hazard and risk analysis
P20 = Organization functional context
P21 = Organization system element
P22 = Organization environment element
P23 = Organization behavior modeling
P24 = Organization mode
P25 = Organization functional structure
P26 = Organization functional behavior
P27 = Organization hazard and risk analysis

Implementation Analysis Places
P28 = Product implementation context
P29 = Product physical external interface requirements
P30 = Product implementation architecture interconnects
P31 = Product implementation architecture flows
P32 = Organization implementation context
P33 = Organization physical external interface requirements
P34 = Organization implementation architecture interconnects
P35 = Organization implementation architecture flows

Loops Places
P36 = System architecture
P37 = Satisfaction loop
P38 =Product requirements loop
P39 =Organization requirements loop
P40 =Product design loop
P41 =Organization design loop
P42 =Product verification loop
P43 =Organization verification loop
P44 =Product satisfaction loop
P45 =Organization satisfaction loop

Figure 6. Description of the places in the Petri Nets graph for the Systems Concurrent Engineering process



System Mission Transitions
t0 = Identify the product mission
t1 = Identify the product life cycle process
t2 = Identify their scenarios and scenarios with the scope of the development effort

Stakeholders Analysis Transitions
t3 = Identify product stakeholders and their concerns
t4 = identify organization stakeholders concerns and their concerns
t5 = Identify product stakeholders requirements and define product measures of effectiveness
t6 = Identify organizations stakeholders requirements and define organization measures of effectiveness

Requirements Analysis Transitions
t7 = Transform product stakeholders requirements in technical requirements
t8 = Transform organizations stakeholders requirements in technical requirements

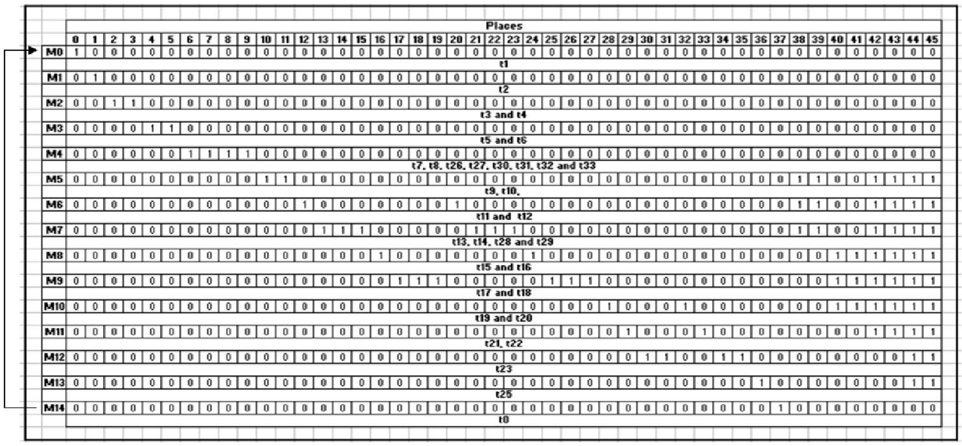
Functional Analysis Transitions
t9 = Identify product functional context
t10 = Identify organization functional context
t11 = Define product system element, identify product environment element and product behavior modeling
t12 = Define organization system element, identify organization environment element and organization behavior modeling
t13 = Identify product modes
t14 = Identify organization modes
t15 = Define product functional attributes
t16 = Define organization functional attributes

Implementation Analysis Transitions
t17 = Identify product implementation context
t18 = Identify organization implementation context
t19 = Define product physical external interface requirements
t20 = Define organization physical external interface requirements
t21 = Define product implementation attributes
t22 = Define organization implementation attributes

Loops Transitions
t24 = Define system architecture
t25 = Identify satisfaction loop
t26 = Identify product requirements loop
t27 = Identify organization requirements loop
t28 = Identify product design loop
t29 = Identify organization design loop
t30 = Identify product verification loop
t31 = Identify organization verification loop
t32 = Identify product satisfaction loop
t33 = Identify organization satisfaction loop

Figure 7. Description of the transitions in the Petri Nets graph for the Systems Concurrent Engineering process

Figure 6 and 7 present the semantic formalism that describes the function of places and transitions in the Petri Nets Systems Concurrent Engineering process generic model. From the Petri Nets graph, it can be applied the Petri Net analysis tools. For example, Figure 8 presents the reachability tree. The reachability tree is basic to study the dynamic properties of any system modeled with Petri Nets. The triggered transition modifies the distribution of marks or tokens on the graph of Petri Nets. In the definition of Petri Nets, it is called reachability of a mark *Mn* the set of all the markings generated from *M0*.



	Places																																												
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
M0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	t1																																												
M1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	t2																																												
M2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	t3 and t4																																												
M3	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	t5 and t6																																												
M4	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	t7, t8, t26, t27, t30, t31, t32 and t33																																												
M5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1
	t9, t10,																																												
M6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1
	t11 and t12																																												
M7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1
	t13, t14, t28 and t29																																												
M8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	t15 and t16																																												
M9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	t17 and t18																																												
M10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
	t19 and t20																																												
M11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1
	t21, t22																																												
M12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1
	t23																																												
M13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1
	t25																																												
M14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	t0																																												

Figure 8. Reachability of the Petri Nets graph for Systems Concurrent Engineering process

5 Conclusion

The generic model is represented by graphical tools and semantics formalism of Petri Nets that allows a visualization of high abstraction of the concurrent activities for integrated development of complex products by Systems Concurrent Engineering process. Dynamic analysis, simulation, verification, implementation and design analysis of iteration throughout the integrated development process can be analyzed by graph of Petri Net, for example using a reachability tree. From the generic model proposed, it is possible to develop models specific to a domain of application including, for example, the various decision making points during the complex product architecting process.

For a space satellite development, for example, decisions to be made are: which stakeholders to satisfy, which requirements to meet, which concept to choose along the life cycle processes, which functions the product and organizations shall perform, which alternative reference architecture models to choose, which solutions to choose in order to implement the chosen architecture. Further steps of this work are to demonstrate how to move form the generic model to a given application domain and in that domain develop a tool that anticipates to the early complex product development stages, the choices and decisions, and therefore their consequences. Also, the tool will provide support along the system life cycle process and will incorporate the lessons learned.

This will allow a gain in productivity in the system architecting process, will allow a common language to be shared among different stakeholders along the system life cycle process and will allow focus of product development in alternative solutions of greater potential.

6 References

[1] Bandecchi M. The ESA Concurrent Design Facility (CDF): concurrent engineering applied to space mission assessments. ESA - 2nd Nordic Systems Engineering Boat Seminar. ESA/ESTEC Noordwijk, NED, 2001.

[2] Bandecchi M, Melton B, Ongaro F. Concurrent engineering applied to space mission assessment and design. ESA bulletin, NED, 1999.

[3] Garbi GP. Application of Petri Nets in System Identification and transport of three different types of products. Master in Engineering Thesis, Taubaté, BRA, 2004.

[4] Garbi GP, Grandinetti FJ. Intelligent Transporter System Modeling by Petri Nets. 24º International Congress of the Aeronautical Sciences - ICAS, JPN, 2004.

[5] Girault C, Valk R. Petri Nets for Systems Engineering. A Guide to modeling, verification, and applications. Berlin Heidelberg New York: Springer, 2002; 473–539.

[6] Koo HYB. A Meta-language for Systems Architecting. PhD Thesis. Massachusetts Institute of Technology, Massachusetts, USA, 2005.

[7] Loureiro G. A systems engineering and concurrent engineering framework for the integrated development of complex products. PhD Thesis. Loughborough University, Loughborough, UK, 1999.

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

[9] Systems Engineering Handbook. International Council on Systems Engineering - INCOSE, USA, 2010.

[10] Wall SD. Use of concurrent engineering in space mission design. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, USA, 1999.

[11] Zhou J, Carmi S; Lau A, Koulas S. Concurrent engineering concepts applied to concurrent education and research. Third ISPE International Conference on Concurrent Engineering. Ontario, CAN, 1996.