INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKOLM, AUGUST 19-21, 2003

SPECIFICATION AND MODELLING OF HYBRID SYSTEMS: AN UML AND PETRI NETS INTEGRATED CONTRIBUTION

Faiçal Miled, Nicolae Brinzei, and Michel Ferney

Abstract

Nowadays, the unceasingly demand for innovative functions, at reasonable costs and high performances, involves an increased complexity in designing product. Indeed, the multifunction hybrid systems prove of a considerable complexity at the stage of the design. In order to overcome this problem a detailed specification and a semantic bases of all the functionalities of the system to be designed are required. A complete and formalized description of the system allows the clearness of the specification, decreases contradictions and increases information density necessary to the conceptual phase of an engineering design procedure. In this paper we both use the Unified Modelling Language (UML) diagrams for the specification and functional modelling of a hybrid systems and the Petri Net tool for the synthesis and the validation of the control part. These tools allow a strong communication between all the design actors and a generic approach for specifying and validation of hybrid systems functional specifications.

Keywords: mechatronics, systems engineering, functional modelling, cooperative tools.

1. Introduction

In this paper we present a generic approach for specification and modelling of a hybrid systems class transmitting power between sources and a load.

In order to face the unceasingly complexity of systems and to manage the various phases of their development, a detailed specification and a functional design model of these systems and their environment are required. Various tools of modelling exist, for example, SADT method (Structured Analysis and Design Technique) and QFD method (Quality Function Deployment) [1]. SADT Method proposes a hierarchical modelling which obliges to order the data and the activities by composing or recomposing them. QFD method is adapted to the redesign problems. It is presented in the form of a method of differentiation, alternatives traceability or capitalization of knowledge. In contrast, the initialisation of QFD method on a new project can be long. It can also involve difficult situations. In order to mitigate these disadvantages, we propose the use of an object-oriented design approach. For several reasons, this approach knew a great success for modelling systems [2]: a high abstraction degree, obtaining more compact models, simpler and less sensitive to the future changes in the real corresponding systems. This approach largely proved reliable for modelling of the information systems. We introduce it, in our case, at various stages of the step of hybrid systems design. Among the tools developed on this approach, we propose the UML formalism [3], which succeeds to a visual modelling of the structure and system behaviour. The first part of this paper deals with the hybrid systems and their environments. The second part treats the use of a qualitative and quantitative approach reasoning to start specification. The last part cops with our approach of specification and modelling of the hybrid systems using UML and a systematic passage between UML statechart diagram and Petri Net Model in order to synthesize and validate the active part of hybrid system.

2. GENERAL INFORMATIONS ON THE SYSTEMS

2.1 Technical system

In accordance with the completeness law of TRIZ method [4], a technical system contains four principal elements: a Source (effort or flow) Unit (SU), a Transmission Unit (TU), a Working Unit (WU), which represents the load and a Control Unit (CU) figure1. The SU provides energy conversion necessary to the realization of the use functions. The TU ensures the transfer of energy between the SU and the WU. The CU supervises and controls all the rest. The architecture of a technical system is given by figure 1. To each technical system we associate a triplet τ (SU, TU, WU) from which results the physical model of its passive part, which represents an interconnection between SU, TU and the WU. The CU represents its active part.

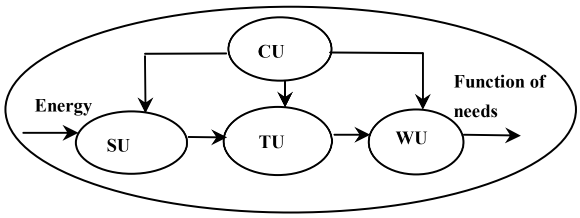

Figure 1. Architecture of a technical system.

2.2 Hybrid system

A hybrid system can have a discrete and continuous dynamic behaviour. It can have many configurations of technical systems. This is due to the possibility of having many SUs (ex. hybrid vehicle). Some changes of operating mode induce change of the configuration of the triplets, which characterizes the technical systems that induces a discontinuity of the physical meta-model of the hybrid system. This discontinuity is due to a process of disconnection and connection of one or several elements (SU and/or TU) vis a vis the WU. This process is ensured by a system of transitions TRS. That induces a creation of another technical system, which must carry out another principal functionality. A hybrid system (figure 2) can then have several technical systems. Each one has a continuous dynamic behaviour. In our approach, the elements concerned with the specification and thereafter the design, are TU and CU. Together, they represent the Set to be Designed (SD). To each technical system, we associate a triplet τ_i (SU_i, TU_i, WU) supervised by a CU_i. Other than SD_i, the TRS which ensures the discrete dynamic behaviour, will make it also the object of specification and thereafter of the design. The active part of a hybrid system is constituted by the TRS and all the CUs of the possible technical systems.

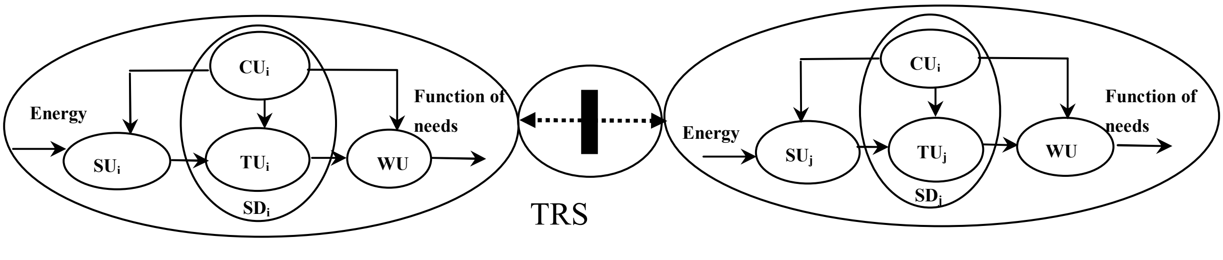

Figure 2. Architecture of a hybrid system.

The hybrid system meta-model is a discontinuous model, composed by several models. Each one relates to the model of the passive part of a technical system. The design of the TRS depends only on technological criteria, however the synthesis of CUs, depends on control criteria. The model of such a triplet τ_i (for instance states equation, transfer function or bond-graph model) must obey to control criteria. These criteria are fundamental in the selection stage of TU_i candidate solution. The existing approaches [1] ignore these criteria. They are often based on dimensioning criteria, whose validation is done by many simulations steps with different value sets. This generates under optimal design, and more difficulties in design CU_i. In this stage of design, it is interesting to isolate all possible triplets early, in particular at the specification stage. In our approach we consider that SUs and the WU are generally preset. It remains to design the TUs by taking into account the control criteria, such as the controllability, observability and invertibility [6], which constitute prerequisites for the design of various CUs.

3. SPECIFICATION

For a hybrid system, the specifications of the TU and CU of each technical system are independent. We consider in our approach that technical systems ensure independent principal functions. All these functions as well as the function of TRS represent the global function of the hybrid system. The step of specification is based on state-transitions approach. We can distinguish two abstraction levels. A higher level of abstraction, that consists in describing the various technical systems that a hybrid system can have. The second level consists in describing the desired functionalities of each technical system. In this case, a technical system will be described by an operating mode on a general abstraction level. A technical system must also operate in more specific operating modes. The TU of one technical system represents a means of an appropriate adaptation of WU and SU power components. We affect states and transitions to the SU, TU and WU. The states of SU are deduced starting from their power characteristics. However, the states of the WU are specified by the designers. The WU represents the element more in relation with the needs; its states represent a specific expression of the functions of needs. A state of a TU ensures a correspondence between a WU state and SU state. The transitions represent the means of passage from one state to another. The designers must evaluate on the transition possibility and on its nature (continuous or discrete) in the context where it is desired.

3.1 Qualitative reasoning

The purpose of a qualitative reasoning is to provide a physical representation of a system making it easier to understand and predict its behaviour. This technique was explored in design process [5]. It explains the behaviour of the system by using a symbolic vocabulary, precise and simple, rather than a precise numerical model, unattainable at this stage of specification. In the case of hybrid system, the functional description of each technical system which constitutes it, is represented by a qualitative state QS_i (for i:=1 to n; with n represents the number of qualitative states). The object of this reasoning is the identification of all the technical systems that a hybrid system can have. This makes it possible to synthesize discrete dynamic behaviour.

3.2 Quantitative reasoning

The quantitative states represent desired specifications, more precisely than qualitative states. They can represent quantitative specification of a power variable or a parameter derived from this one. They can also represent an operating point or a particular characteristic to attain. Some examples: acceleration, maximum speed, or maximum engine efficiency...

Each technical system is represented by a QS_i, the designers must establish the quantitative states of each element of the correspondent triplet τ_i (SU_i, TU_i, WU). We note by:

- QSU: a quantitative state of the SU;
- QWU: a quantitative state of WU;
- QTU: a quantitative state of TU, it represents a correspondence between a quantitative state of SU and a quantitative state of WU. The designers must validate this correspondence. A TU_i acts simultaneously on SU_i and WU via its CU_i in order to realize this correspondence.

4. UML modelling

In order to model a hybrid system, we were interested in UML formalism (Unified Modelling Language). UML [3], is a language of object oriented modelling which almost became a standard in the object approach. This language was developed and used for the modelling of the information systems. Recently, the language was integrated in the design of general products, in particularly, the mechatronic products [7]. UML makes it possible to model in a clear way and specifies the structure and the system behaviour. UML modelling is done by the means of the diagrams defined in UML standard. In order to represent the structure of the hybrid system, we use the UML class diagrams. The use case diagrams describe the functional behaviour of the system as seen by an external user. The collaboration diagrams show how objects associate with each other. And finally, the statechart diagrams models dynamic behaviour of classes or objects.

4.1 Use case diagram

The use case diagram represents the starting point of UML modelling. It describes the functional behaviour of the system as seen by user. For the hybrid systems, we represent, in a first level, the number m of the qualitative states QS_i (for i:=1 to m). These qualitative states will be assigned to use cases UC_i. Each use case is supposed to be split into several extend use cases EUC. If it is the case, the quantitative states $QTU_{i,j}$ resulting from a qualitative state QS_i will be assigned to the $EUC_{i,j}$ We consider that the qualitative state QS_i, is split into n quantitative states. In occurrence a use case UC_i is split into n EUCs. This quantitative state number n is deduced from the possible correspondences validated by designers between a p number of SU quantitative states and a q number of WU quantitative states. We note by $EUC_{i,j}$ (for j:=1 to n) the EUCs of a UC_i. The diagram of figure 3 illustrates a generic use case diagram for hybrid system.

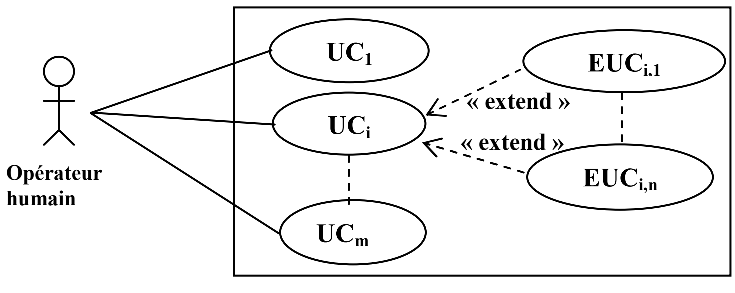

Figure 3. Use case diagram of a hybrid system.

4.2 Class diagram

The class diagram expresses in a general way the internal structure of the system, in term of classes and relations between these classes. A hybrid system is represented by a meta-class «Hybrid_System» consisted by «Technical_System» super-classes coexistent with «Transi» super-class. Each super-class «Technical_System» ensures the realization of one UC by the technical system that it models. The super-class «Transi» ensures the discrete transitions between these UCs. The super-classes «Technical_System» are also constituted by three classes which are «Source», «Transmission» and «load» coexistent with a «Control» class which ensures the realization of the EUCs. We also define, for each class, the attributes and the operations taken into account in modelling. Figure 4 represents the class structure of the meta-class «Hybrid_System» composed by m super-classes «Technical_System» and a «Transi» class. For a hybrid system, other classes are also associated to the meta-class «Hybrid_System». In fact the external environment diagram provides the definition of the classes: «Human_Operator», «External_Sollicitations» which coexist with the meta-class «Hybrid_System». The class «External_Sollicitations» gathers and characterizes the external environment. It makes it possible to highlight the attributes and the operations in relation with the internal behaviour of the classes with which it is in connection. We can quote the following examples: soil, the gravity field, ambient conditions and specific external mediums. Human operator, which represents a modulated input, coexists with the active part of the hybrid system. His instructions are translated into operations.

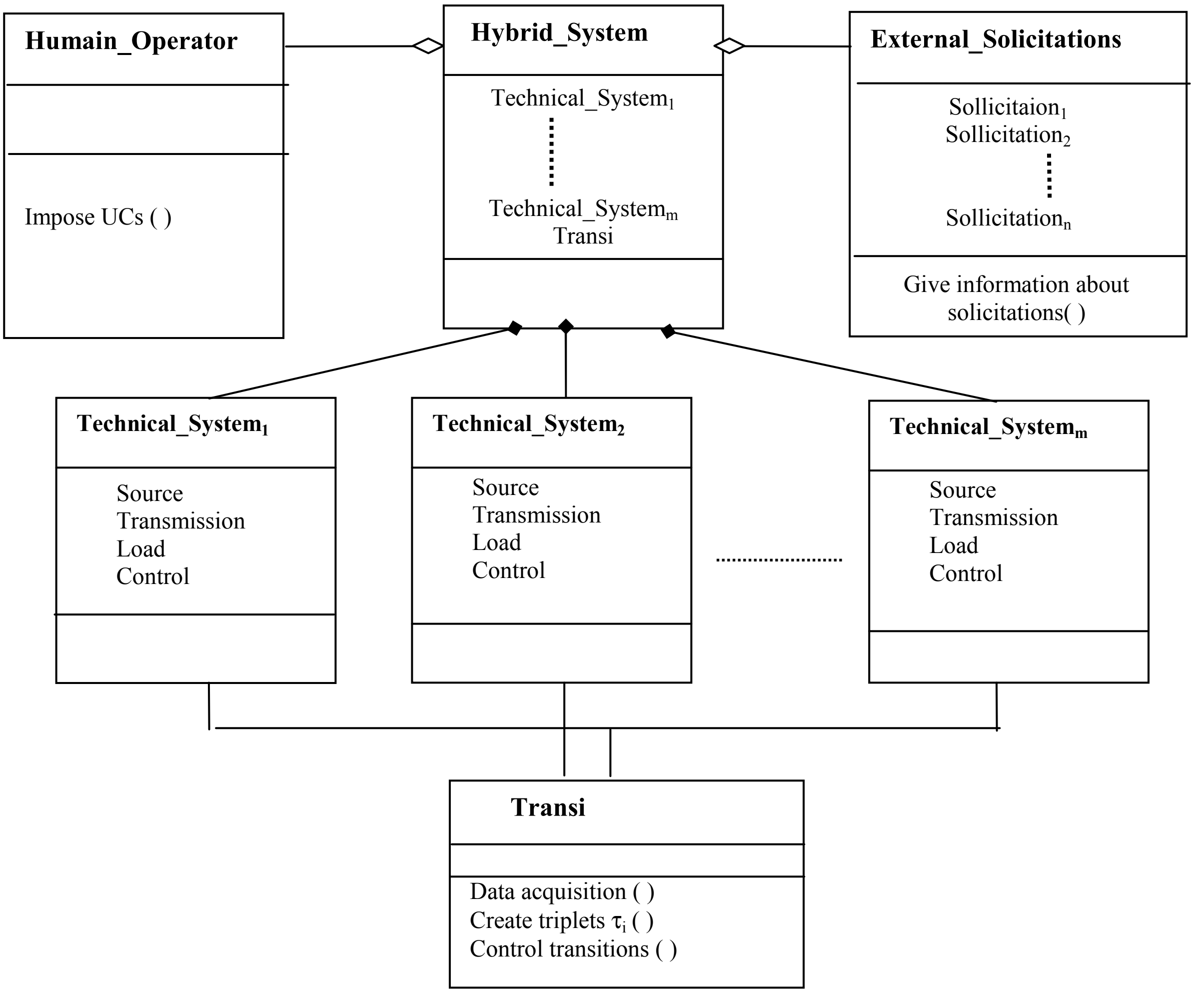

Figure 4. Classes Diagram of a hybrid system.

Each «Technical_System» super-class must carry out one use case UC. The objective is to realize the QWUs, which are related to the needs functions. Each QWU_i represents an operation to be carried out in the «Load» class by the intermediate of two quantitative states QSU_j and QWU_k. They represent respectively two operations in «Source» and «Transmission» classes via the operations Realize QWU, Realize QTU and Realise QSU in «Control» class.

Figure 5, represents a structure of one «Technical_System» super-class.

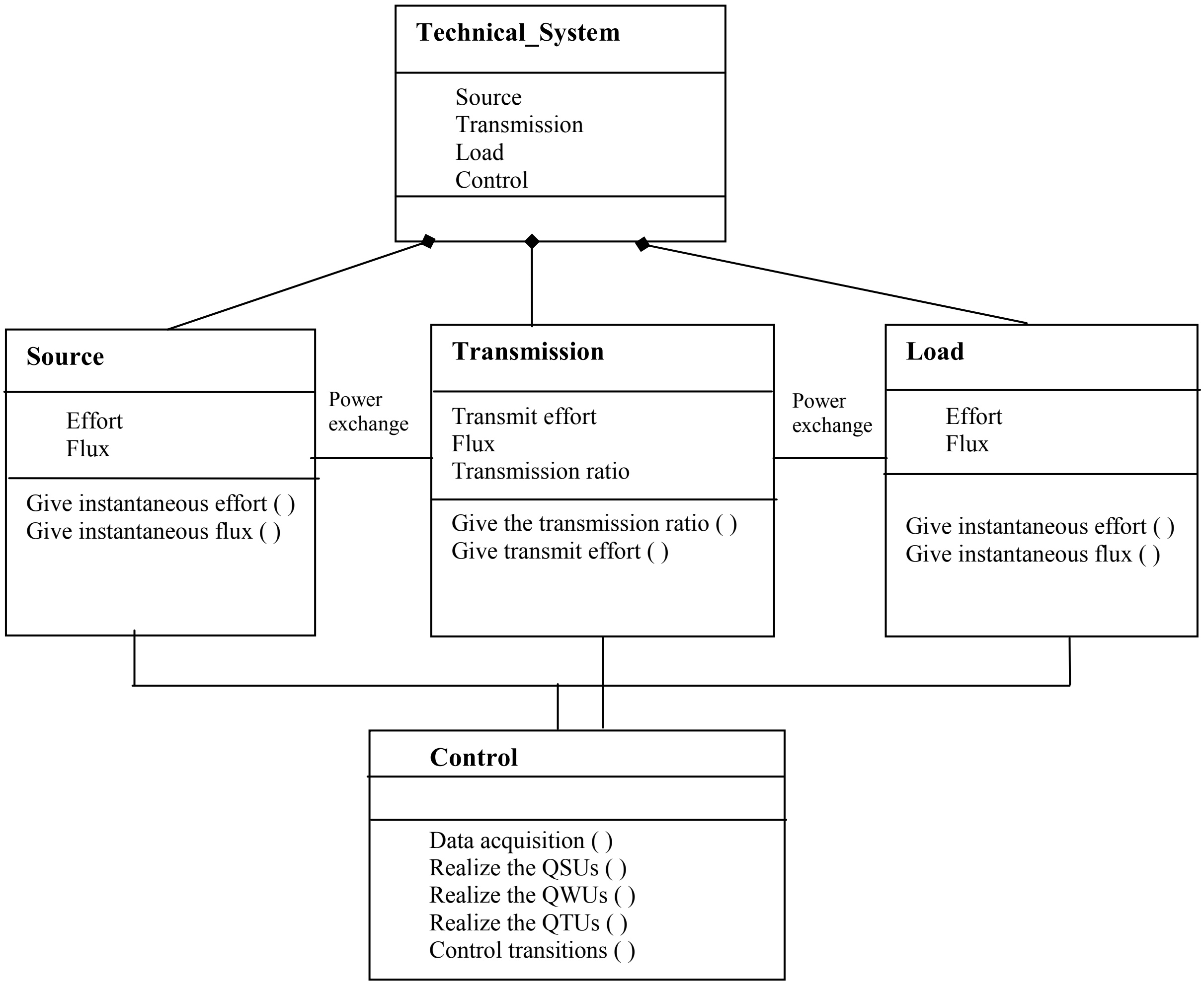

Figure 5. « $Technical\_System_i$ » super_class diagram.

4.3 Collaboration diagram of a hybrid system

The collaboration diagram describes the roles of objects instance of classes in the context of the realization of the EUCs. In our approach, we consider that the SU and WU objects are preset. Therefore they constitute objects instance respectively of «source» and «load» classes. However, the designers are asked to design objects instances of «Transmission», «Control» and «Transi» classes. Each object is specified in order to realize the UCs in general and more particularly the EUCs (qualitative and quantitative states). We suppose that a UC has n EUCs in occurrence n quantitative states QTU. We also suppose that we have n QSUs and n QWUs. These correspondences are represented in figure 6: the collaboration diagram. In order to represent the collaboration diagram, we assign roles to a SU object instance of «source» class, TU object instance of «transmission» class and to a WU object instance of «load» class. In fact these roles represent the quantitative states, which must ensure the correspondent objects via the CU object instance of «control» class. Figure 6 represents the collaboration diagram of one EUC.

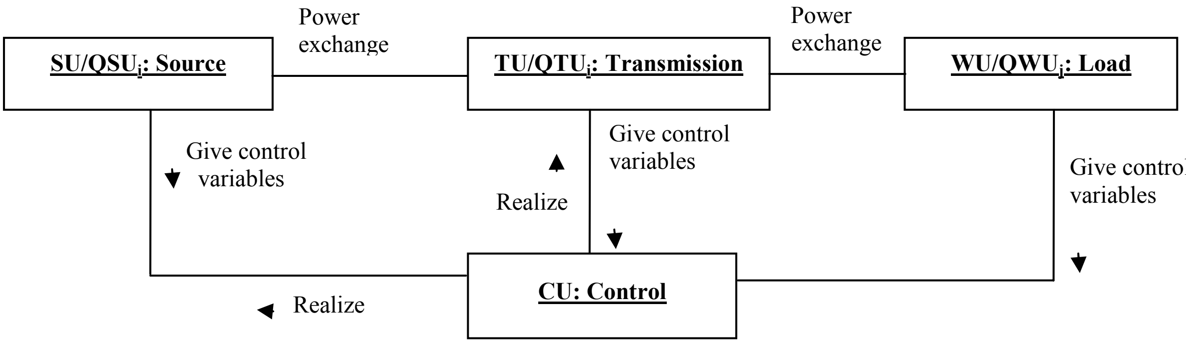

Figure 6. Collaboration diagram.

4.4 Statechart diagram

Each object is for instance in one particular state. The evolution of the object in this state and its collaborations with the other objects determines the transitions. UML formalism proposes the statechart diagrams. The SU and the WU objects are preset and their possible states were defined during the phase of specification in the quantitative reasoning step. These states and the transitions between these states are modelled by statechart diagram (figures 7.a and 7.b). The statechart diagram of one TU object is represented in figure 7.c. The QTUs represent really the EUCs relating to one UC, that our system must realize. The transitions between these states modelled in the statechart diagram must be ensured by the CU object.

For each technical system belonging to the hybrid system, we develop the statechart diagrams of SU, TU and WU, in order to know their internal behaviours.

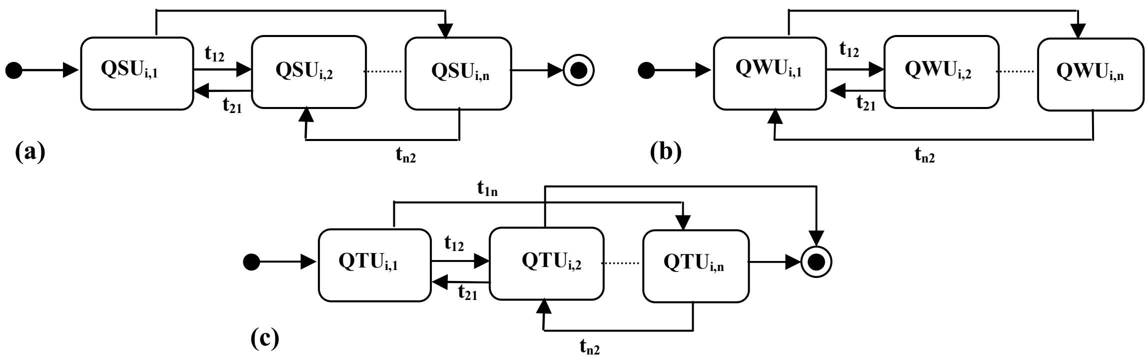

Figure 7. Statechart diagram of technical system elements.

The statechart diagrams are generally used to model classes behaviour, but they can also represent the dynamic aspects of others UML modelling elements. Thus, we propose its use for modelling the dynamic aspects of use case diagram, the transitions to be realized in possible UCs (figure 8).

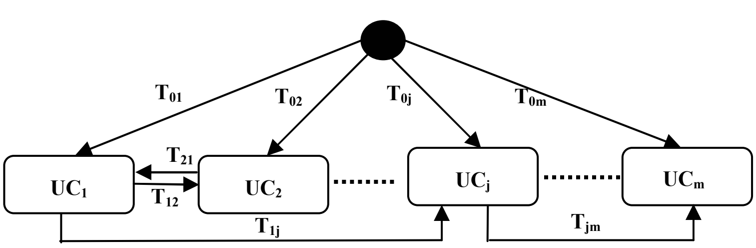

Figure 8. Representative statechart diagram of UCs dynamic aspect.

Until now, we have modelled the hybrid system by using UML formalism. We thus obtained a pragmatic approach for modelling these systems, which makes it possible the clarification of the task stage in design process. We have also taken into account the structure of the system and of its behaviour. Those are modelled in various used UML diagrams.

Let's now move to the second objective: synthesis and validation of the desired behaviour.

5. Synthesis of the hybrid system active part

5.1 Petri Net model of the hybrid system active part

The control of the hybrid system is ensured by his active part: the CUs of technical systems and TRS. In order to determine and to model the active part, we propose the use of Petri Nets [8]. The Petri Net is a modelling tool, which fits well to the description of the systems in which problems of synchronism and parallelism occur. This tool also allows a hierarchical modelling at various levels of operating system. TRS ensures the transitions between the different technical systems. Each technical system is carrying out a general use case described in the diagram of figure 3. To each UC_i, We affect a place in the Petri Net. We add an additional place representing the technical system in initial state awaiting an instruction from a human operator. The transitions between these states are obtained from the transitions between UCs of statechart diagram (figure 8). The Petri Net representing the behaviour of TRS is given in figure 9.

The places of the Petri Net represented in figure 9 describes the UCs and some ones they can be split into different EUCs in accordance with the diagram of figure 3. Within the meaning of the Petri Nets, these places are called substitution places associated to a sub-Petri Net model with lower hierarchical level. In order to describe the detail of the associated behaviour with this type of places, we use the TU statechart diagram (figure 7.c). This diagram represents the various EUCs and the transitions between them. In each state of the statechart diagram of the figure 7.c we make a correspond place of the sub-Petri Net and with each transition between the states we make a correspond transition of the sub-Petri Net. The sub-Petri Net obtained is given in figure 10. The Petri Net of the figure 9 represents the behaviour of the TRS. The sub-Petri Net of the figure 10 represents the behaviour of one hybrid system CU related to one UC. By integrating the various CUs sub-Petri Net models of each UCs in TRS Petri Net model (figure 9), we obtain a total Petri Net describing the operating of hybrid system. We will proceed now to the validation of this operating.

5.2 Validation

The obtained Petri Net model allows the simulation of the hybrid system. The simulations of diverse UCs necessitate knowledge of TUs candidate solutions models, SUs and WU models. These models will be represented by bond-graph tool [6]. Bond-graph allows principally the taking into account of structural properties such as the controllability, observability and invertibility early, i.e. at the passive part design stage and system simulations.

At this stage it is very interesting to carry out a structural analysis of the Petri Net model, which informs us about the completeness of our study and the validation of the preliminary solutions. Thus, we can study the vivacity of the Petri Net model. The vivacity property of a Petri Net [8] guarantees the firing of all transitions whatever the initial Petri Net markings is. It represents the most important Petri Net property. It helps to verify that any place attained doesn't represent a blocking state. Another interesting property consists in checking the accessibility of all Petri Net places, which means that all the states of the hybrid system are attainable. If these properties of Petri Net are checked out, we can conclude on the completeness of analyse and the validation of requirements specifications.

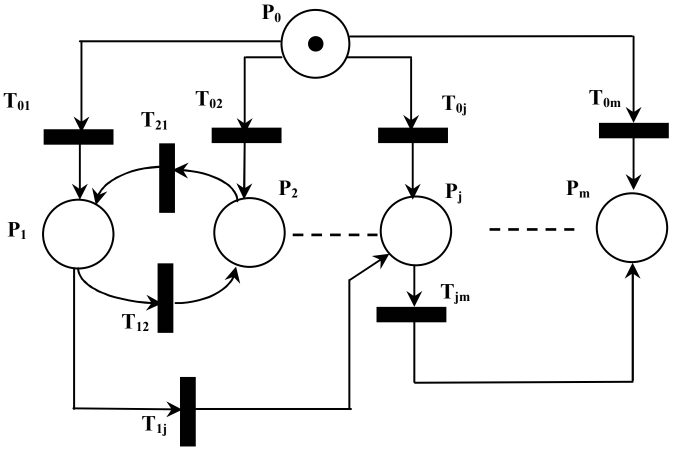

Figure 9. TRS Petri Net model.

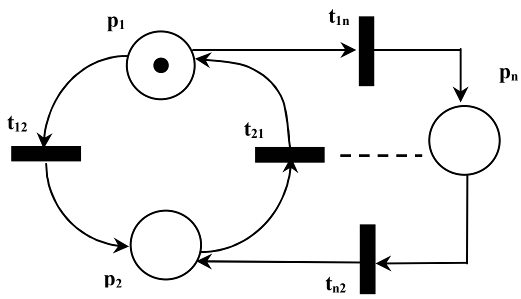

Figure 10. CU sub-Petri Net model.

6. Conclusions and prospects

In this paper, a generic approach for valid hybrid systems functional specifications using UML and Petri Nets was proposed. This work was applied to automatic transmission of scooter [7] and in this paper we generalize for all hybrid systems.

An advantage of UML is revealing of gaps and inconsistencies in requirements specifications on very early stage of design, as well as ease of understanding and modification of visual modelling diagrams. Unification and precision of notation is important for large and interdisciplinary projects like the design of hybrid systems. UML keep track of used class and object names, its attributes and methods. Designers may transfer already classes and other elements between different diagrams and reuse them. This accelerates work, progress and helps to keep all parts of project consistent.

The passage from UML diagrams to Petri Net model allows studying the active part by verifying essentially the vivacity of Petri Net model and then the validation of the active part. The Petri Net represents also a simulator for the system. Model simulation needs SUs, WU models and TUs technological solutions models, in fact model of hybrid system passive part. Models of TUs technological solutions represent a tool for the realization of all system states by an appropriate adaptation of SUs and WU power components. These models will be represented by bond-graph tool, which represents an intermediary between functional representation and behavioural model. The generation of TUs alternatives design should take into account control criteria, which are verified directly on bond-graph model [6]. The Petri Net also ensures the supervision and the control of various bond-graph models. Petri Net and bond-graph models form a complete virtual prototype of the hybrid system. This prototype also makes possible to carry out qualitative, then quantitative simulations in order to compare the results with the specifications [5], without referring to technological solutions.

References

[1] Tollenaere E. ″Conception de produits mécaniques – méthodes, modèles et outils″, Editions Hermès, France, N° ISBN 2-86601-694-7, 1998.

[2] Muller P.A., Gaertner N. ″Modélisation objet avec UML″, Editions Eyrolles, France, N° ISBN 2-212-09122-2, 2000.

[3] OMG "Unified Modeling Language Specification", version 1.3, 1999.

[4] Altchuller G. "TRIZ The innovation algorithm; systematic innovation and technical creativity", technical Innovation Center Inc., Worcester, 1999;

[5] Yannou B., Vasiliu A. "Design platform for planar mechanisms based on a qualitative kinematics, International Workshop on Qualitative reasoning about physique Systems" QR'95, Amsterdam, PP 18-19, 1995.

[6] Dauphin-Tangy G. "Les bond-graphs" Hermès science, France, Publication 2000;

[7] Brinzei N., Ferney M. and Miled F. "Modélisation UML pour la conception des systèmes mécatroniques: application à une transmission automatique" 4th international conference on integrated design and manufacturing in mechanical engineering IDMME 2002, CDROM.

[8] David R., Alla H. "Du Grafcet aux réseaux de Petri", Hermès, France, N° ISBN 2-86601-325-5, 1997.

For more information please contact:

Michel Ferney Université de Technologie de Belfort-Montbéliard - 90010 Belfort Cedex (France)
Tel: 03 84 58 33 40 – Fax: 03 84 58 31 46 E-mail: michel.ferney@utbm.fr