# Multidisciplinary Simulation Model Development: Early inconsistency detection during the design stage

Göknur Sirin, Fabien Retho, Bernard Yannou, Martine Callot, Philippe Dessante, Eric Landel

Sirin G., Retho F., Yannou B., Callot M., Dessante P., Landel E., 2017, "Multidisciplinary Simulation Model Development: Early inconsistency detection during the design stage", *Advances in Engineering Software*.

# Multidisciplinary Simulation Model Development: Early inconsistency detection during the design stage

Göknur Sirin[1,4], Fabien Retho[2,3], Bernard Yannou[1], Martine Callot[3], Philippe Dessante[2], Eric Landel[4]

[1] CentraleSupélec, Laboratoire Génie Industriel, Chatenay-Malabry, France
[2] Département Energie Supélec, Gif-sur-Yvette, France
[3] Airbus Group Innovations, France
[4] Renault SAS Techno-Centre, Guyancourt, France

## Abstract:

Integration, Verification and Validation (IVV) practices in simulation-based design helps reduce inconsistencies in multidisciplinary systems, i.e. those combining multiple mechanics, structural, hydrodynamic or other complex components. In current multidisciplinary simulation model development processes, sub-system simulation models are usually Verified and Validated (V&V) at supplier level to assess whether a system meets its intended goals; in such scenarios, each system is verified and validated separately – mechanical, structural, hydrodynamic, etc. However, many problems may arise during the actual integration of these modular sub-system simulations at the Original Equipment Manufacturer (OEM) level, which increases the risk of late inconsistencies such as interface mismatches or other interoperability-based problems. To address this problem, the present work aims to reduce late inconsistency detection through ensuring early stage collaborations between the different suppliers and the OEM by proposing a clear simulation model request. Our approach is illustrated with an industrial case study showing how a Model Request Package that contains the Model Identity Card (MIC) and Model of Intention (MoI) concepts, helps reduce the knowledge gap and inconsistencies between OEMs and model suppliers.

**Keywords:** Collaboration, Multidisciplinary Modeling & Simulation, Model of Intention, Model-Based Systems Engineering, Ontological Engineering, Consistency Management

---

## 1. Introduction

To represent vehicle behavior accurately in a given field situation, simulation models are created by integrating different simulation models of the sub-systems within the vehicle, such as chassis, engine, transmission, etc. These sub- modular system models are referred to as domain models. The simulation model development process involves a number of parallel or/and sequential activities in which experts in different disciplines create or reuse domain-level models to construct a full vehicle model [1], [2]. This is particularly challenging for the design of multidisciplinary systems in which components in different disciplines (e.g., mechanics, structural dynamics, hydrodynamics, heat conduction, fluid flow, chemistry, or acoustics) are tightly coupled to achieve optimal system performance [2]. Each disciplinary model can be developed and evolve following its own semantics and development tools at different rates [3]. Several problems were thus observed during a preliminary study that the authors conducted in the simulation model development practices of both the automotive and the aeronautic industries [1], [35]. From these observations two major worlds were identified: (a) the system world, including the Systems Architect and the Technology Provider for all types of contracting and innovation issues, and (b) the model or physical world including the Model Architect and the Model Provider for model design and development (see Fig. 1).

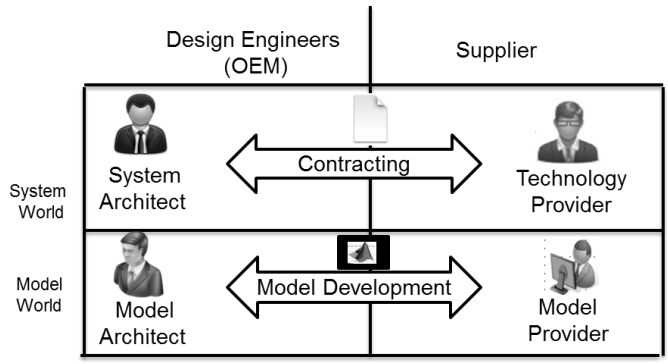


Fig. 1. Collaboration between major M&S actors

Today one of the problems in this collaborative environment is that models are usually considered as available, ready to be integrated or requested directly from an expert (e.g. model provider) without specifying any method. However, the technology or service provider's engineering analysis model is most often in a Black Box (BB) format to preserve intellectual property (see **Erreur ! Source du renvoi introuvable.**1). BB models can be interacted with only through the inputs and outputs of a well-defined interface. The challenge in using a BB model is taking into consideration the number of distinct interface issues, parameters, or messages that have to be passed among the components [4]. Another problem is that the supplier is not integrated in the model design and specification phase, and the OEM has no access to the simulation model development activity. This decoupled way of working may create unnecessary iteration during the model integration phase, often resulting in increased product development lead time and cost. In our experience, specifying a simulation model is not as trivial as might be expected; it is an important yet undervalued practice. Hence OEMs must establish an effective methodology, based on different viewpoints of the actors, to create a credible simulation model, while avoiding project delays and unforeseen rework cost by detecting any inconsistency problems before building costly simulation models.

In systems engineering practice, inconsistencies manifest in a variety of forms: violation of well-formedness rules, inconsistencies in redundant information, mismatches between model, their interfaces and test data, and not following heuristics or guidelines. In current practice, most of these inconsistencies are only identified during reviews that are part of the verification & validation activities. In between these reviews there is a possibility of decisions being made based on inconsistent information and knowledge, which can lead to poor outcomes and costly rework. It is impossible to say whether or not a system is fully consistent but at least we would like to show by this paper that some inconsistencies can be detected as early as possible during the design process. Typically, the earlier an inconsistency is identified, the cheaper it is to resolve. A recent paradigm shift in systems engineering known as Model-Based Systems Engineering (MBSE) has the potential for the process of identifying inconsistencies to be performed in an automated fashion. Automated and computer-assisted methods are important enablers for more frequent inconsistency checks and therefore towards continuously verifying & validating systems [7]. Consistency management in MBSE has been studied before [5], [6] and [7]. A review of the related literature reveals that this is still an open challenge and has not yet been investigated at a fundamental level within the context of Model-Based Systems Engineering (MBSE).

Research efforts mentioned in the related literature discuss several methods to check for and potentially resolve inconsistencies that can occur within and across models from different domains. Unfortunately, most of the current work is incomplete and only considers specific types of inconsistencies. This is one of the reasons why some of them do not scale. Detecting inconsistencies outside the scope of logic and mathematics requires additional information to be present in the system. Therefore, additional, computer-interpretable rules

must be defined in the form of, e.g., description logic or meta-models. These must include domain- or application-specific rules. More rules may be necessary to define the relationship between two or more models affecting application or domain-specific properties [11].
The aim of this work was to reduce late inconsistency detection by ensuring early stage collaboration with a clear simulation model request and design artefact negotiation. To this end, this paper proposes an early-stage multidisciplinary simulation model design methodology inspired by Model Based Systems Engineering (MBSE) [24]. Proposed early stage model request package consists of a meta-model called MIC for interface inconsistencies detection and a Model of Intention (MoI); an executable behavioural model to illustrated expected model's behaviour.
This paper is organized as follows: in Section 2, we give a general literature review on the role of different actors and the management of their views and viewpoints; in Section 3, we introduce our methodology of early-phase simulation model design, explaining the formal architecture design, and MIC and MoI concepts (the reader will find in reference [1] and [34] a more detailed state of the art in meta-model and Model of Intention); Section 4 introduces an industrial case study with its validation protocol, and our conclusion and future considerations are in Section 5.

## 2. State of the Art

### 2.1 Related Research Methods and Industrial Projects

The approach proposed in this paper is developed to support the building of a multidisciplinary virtual product (i.e. simulation model). A virtual product is a global model of a future product (car, train, aircraft, helicopter, etc.) that can be simulated to predict its behavior with an estimated accuracy and validity domain. Numerous methodologies that contribute to the virtualization of design with models and simulations have already been developed [24, 25, 26, and 27]. Increasing vehicle complexity requires an ability to manage development decisions proactively, and this can be accomplished through a Systems Engineering RFLP methodology [14].

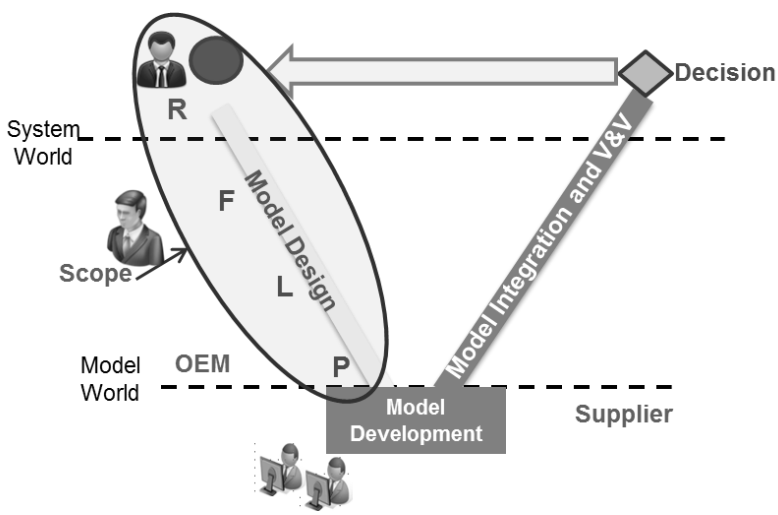


Fig. 2. RFLP Methodology

The RFLP model describes the left-hand descending branch of the "V-Model". Based on the well-known V-cycle design process, RFLP allows concurrent engineering to coordinate the separate activities and views

of distributed design teams. In this context, a viewpoint describes a particular perspective of interest to a set of stakeholders; a view is a stereotyped package said to conform to a particular viewpoint [12] (see Fig. 2).

R refines Requirements views, the Functional View (F) defines what the system does operationally with a given scenario (intended use, purpose, internal functions), the Logical View (L), or logical/organic architecture defines how the system is implemented (i.e. its breakdown structure, block diagram, logical interfaces, and logical connections). The Physical View (P) defines a virtual definition of the real world product. In this paper, we customize these different views based on our needs. Thus a complex simulation model development process can be described in the context of four interrelated types of knowledge – functional, structural, behavioral, and physical knowledge or views. These views contain in turn numerous sub-views; the functional view contains requirement, system architecture and operational views. Structural knowledge includes definitions of the form, and dimensions of the artifact, its constituent components, and their arrangement and connections to each other. A structural description is sufficient to construct the artifact (simulation model). It includes the necessary information about the artifact's explicit parameters, which a designer determines directly in order to generate a physical solution to an abstract problem. Behavioral knowledge includes the description of the artifact's potential behaviors in response to its environment in a given scenario (see Fig. 3).

As vehicle systems are continuously increasing in complexity, it is essential for there to be a process flow to handle the modeling of the vehicle system from beginning to end [2].

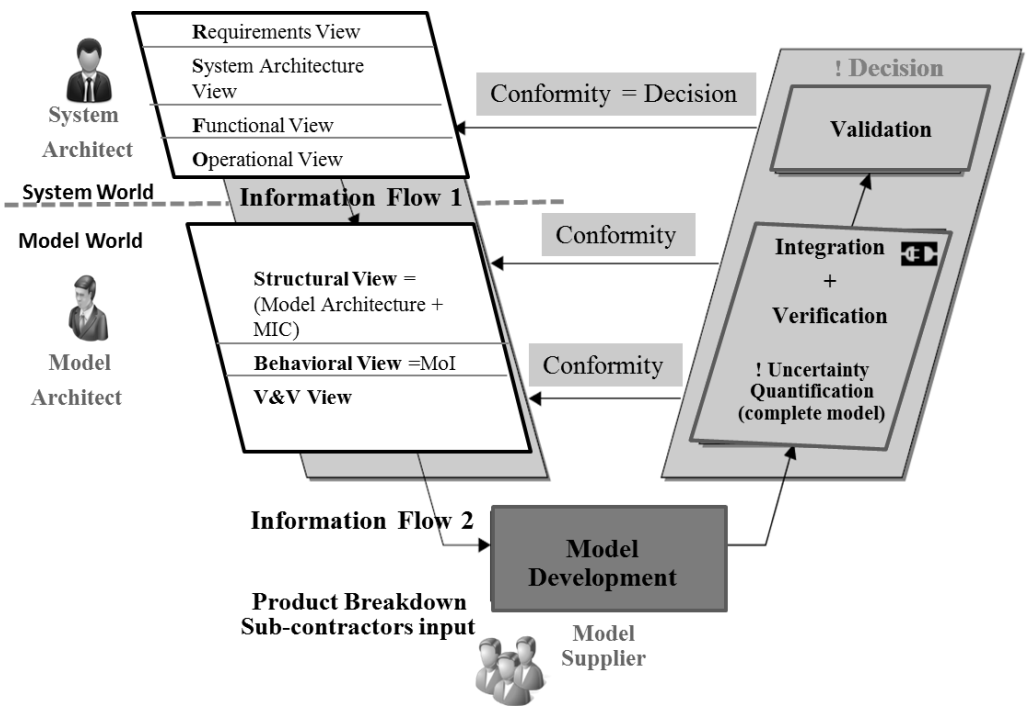


Fig. 3. Conceptual Design Stage

Another industrial methodology called ARCADIA (ARChitecture Analysis & Design Integrated Approach) has been in operational use since 2008 in the Thales Company [37]. ARCADIA is based on architecture-centric and model-driven engineering, and was supported by the use of "standard" language such as architecture Framework (e.g. NATO AF or NAF), and SysML/UML. However, ARCADIA promotes

driving engineering no longer by requirements only, but mainly by functional need analysis (and also operational analysis) [37]. The industrial method stated does not enable us to design a simulation model world. We sought to pull and transform the necessary system-related information such as system requirements, functions and operations to design a simulation model in a model world.

Thus the transition from system models (system world) to behavioral physics-based modeling (model world) should be structured first with an architecture framework, and then supported by information flows between different design actors (see Fig. 3). However, today's internal communication is ensured by informal knowledge sharing, and has become an error-prone activity, because different disciplines often use different vocabularies. Semantic differences can cause misunderstandings among these actors [1].

Today there are two main actors in a model development environment: System Architects and Domain Model Providers (i.e. Model Suppliers). **System Architects** are the sponsors of model development activity. They define the technical and project-based constraints. They also define an operational scenario, some decision criteria, and provide a system architecture. By contrast, **Model Providers** are the domain experts who build models with their specific domain knowledge (see Fig. 3). One of the gaps that we noted during our research investigation in the OEMs is that there is no clear and formal model specification agreement between System Architects and Model Providers at the early model development stage (or conceptual design phase). In addition, most of the time, the interaction of these two actors can create a bottleneck for communication because they do not have the same levels of understanding. One useful result of this work is that it creates a more formal and clearer model request for the domain model providers, so as to obtain the right model at the right time. This transversal view from Functional to Physical View has to be managed by a new actor in the collaboration we name the **"Model Architect"**. Model Architects have a multidisciplinary vision of a product, and simulation knowledge. They also have a deep understanding of both the system-level requirements for the vehicle model, and of how their models have to interface with other domain models (see Fig. 1). There are multiple activities involved in the Model Architecture business cycle, such as:

- understanding the system world requirements, and interpreting and translating these requirements in the model world,
- analyzing or evaluating and selecting the architecture, and
- communicating with system architects and multiple domain-level model providers.

These activities do not take place in a strict sequence, and there are many feedback loops as the multiple stakeholders negotiate among themselves, striving for consensus. The **Domain Expert** (Model Supplier) also plays a fundamental role, with expertise, experience and knowledge to support the Model Provider's final model-building and delivering activities (see Fig. 3).

The collaborative multidisciplinary simulation model development process creates a response to the request of the System Architect who seeks elements to analyze architecture for a design choice. The intent of the System Architect is to evaluate whether system requirements are foreseeably satisfied, based on the existing knowledge of the system to be. During this conceptual design stage, **the first information flow** provides the following elements: an architecture description, variables of interest, constraints, and scenarios with expected accuracy from System Architect to Model Architect (see Fig. 3).

***a. Architecture Description***

The system is functionally and logically described. The logical description covers the topological structure (existing connections among system components, such as SysML Internal Block Diagrams) and other properties specific to the nature of the component (e.g. density, electrical or thermal resistivity, conductivity, chemical nature). When the System Architect considers several alternatives, each one comes with its own architectural description [4, 14].

***b. Decision Criteria***

A variable of interest is a characteristic of the system (related to its behavior, its structure or to any other viewpoint on the system) that interests the System Architect. The variable of interest can be a scalar, a multidimensional vector, or a field in the response space. Noise comfort in a car is one such variable.

***c. Constraints***

***Technical Constraints:***

Unless the System Architect is looking for preliminary or exploratory results, he sets boundaries to surpass (e.g. performance, safety and environment) or to respect (limits) in the response space of variables of interest. The noise level that causes hearing damage is one such limit.

***Project Constraints:***

Project cost and time to development are considered as project constraints.

***d. Scenarios***

Multiple scenarios can be proposed to fit the complexity of the vehicle and the different mission profiles, such as standardized urban, extra-urban, or mixed fuel mileage for a car or in an operational concept (e.g. flight profile typical for a short-range commuting airliner).

***e. Expected Accuracy***

The applicability of any model depends on the accuracy and reliability of its output. However, because all models are imperfect abstractions of reality, and precise input data are rarely available, all output values are subject to inaccuracies. Accuracy is the closeness of the agreement between the measured and true values. High accuracy implies low error. Thus depending on the maturity of the design and design specification supplied to the Model Architect, the System Architect may insist on having a high degree of confidence in the model result produced [10].

**Second information flow** from Model Architect to Model Provider contains the following elements: early-stage simulation model design consisting of a formal architecture representation, an MIC and a MoI. The Model Identity Card (MIC) and Model of intention (MoI), concepts that we will describe in the next section, were developed with the aim of supporting structural and behavioral views of the simulation model design phase (see Fig. 3).

## 3. PROPOSED METHODOLOGY

In this section, we define the MIC and MoI concepts separately, and go on to propose the mixed method.

### 3.1 MODEL IDENTITY CARD (MIC)

Designers and suppliers need to obtain an effective knowledge exchange through a shared vocabulary. Providing a common vocabulary for the design and development actors can help communicate fact-based decisions in a maker's assessment of the credibility of M&S results [7]. The ability for users to select from a list of options is an immensely important capability. By restricting large groups of users to the same vocabulary and set of options whenever possible, inconsistencies arising from miscommunication or misinformation can be reduced significantly [1]. There are some recent, similar research efforts in ontology-based systems engineering. The most extensive prior research on characterizing a model's behaviour in engineering analyses was that of Grosse et al. [16]. This ontology draws upon some of the analysis modeling taxonomies and concepts presented by Noy and McGuinness [17]. They organize the knowledge of engineering analyses models into an ontology, which includes both metadata (e.g. author, documentation and meta-knowledge, such as model idealizations and the corresponding justifications). A similar, though less extensive metamodel for engineering analysis models has been developed by Mocko et al. [18], but these taxonomies do not include detailed model behavior characteristics, or any model validation and verification attributes such as NASA's credibility assessment [19]. Based on the standards and methodologies described above, we developed the MIC metamodel likely to be applicable to any numerical model in the context of vehicle manufacturer with some domain specific customization.

The MIC metamodel includes some important refined characteristics of engineering analysis models (object) such as modeling assumptions and interface specifications.

The objective of MIC is to simplify analysis model specifications, sharing and reducing ambiguity, and to reduce the amount of rework caused by interface inconsistency between domain models.
MIC assigns a model to one of two main classes: an object with four sub-classes (Object, Methods, Usage and Model Quality) includes the attributes of the model, which are stored in the MIC Properties. The second tab (Interface) includes the attributes of the Ports that designers define (see Fig. 4 and Fig. 5).

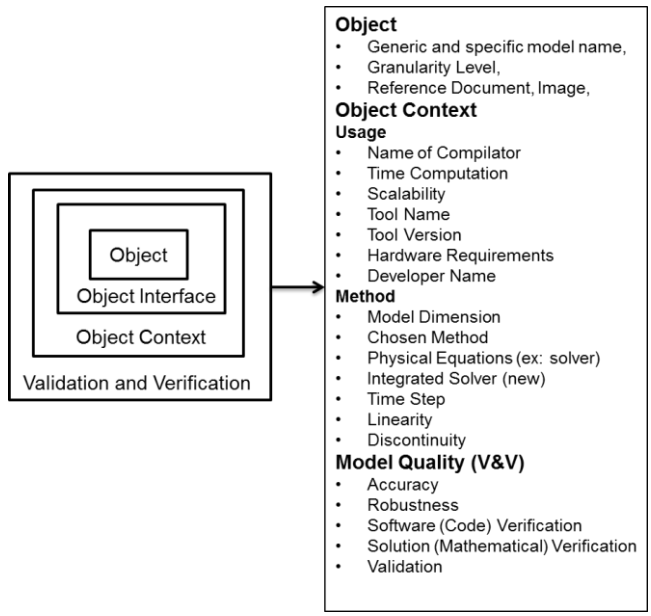


Fig.4. Main Classes and Attributes

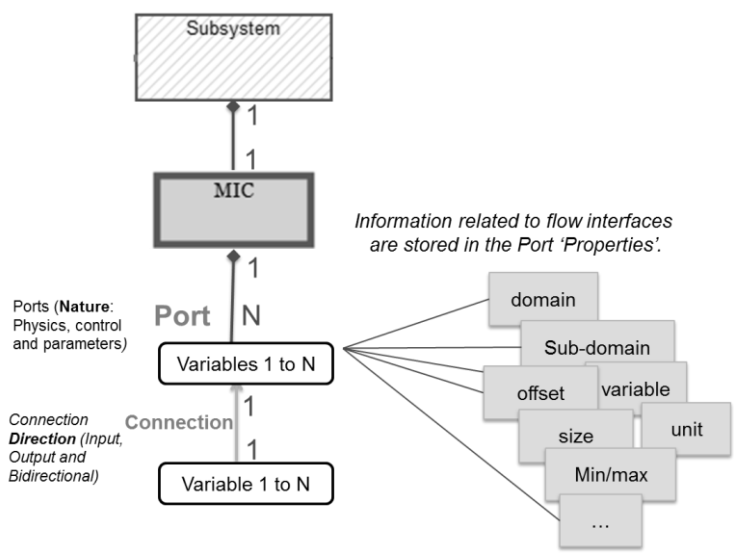


Fig.5. Interface Specification

The semantics and definition of the interfaces need to be accurate and complete because the communications within model components or between model components and the outside environment have to be connected with well-defined interfaces [18]. The interface does not contain any information about the internal behavior of the component. Instead, the interface exposes the key parameters, while encapsulating the implementation of the model, which defines the internal behavior of the component.

The following are some important characteristics of the MIC Interface Concept (see Fig. 5):

- ✓ Each Model, and consequently its MIC, can have several Ports of different Domains (e.g. Mechanical, Electrical, Hydraulic, and Thermal).
- ✓ Each Port is composed of multiple Variables.
- ✓ Variables represent Efforts (voltage, temperature, force, torque, pressure) & Flows (current, entropy, linear velocity, angular velocity, volumetric) based on the law of energy conservation in a physical system [20].

### 3.2 MODEL OF INTENTION (MoI)

The MoI is a model-based approach to request and specify models for a given scenario [34], [35]. It enables a model supplier and technology provider (expert) to propose an adequate model. This concept is innovative in the design field, but comparable in some respects to the System Specification Model (SSM) [31], Simulation Conceptual Model (SCM) [32] and Prescriptive Model [33], because MoI expresses an expected behavior of a simulation model with a specific estimated role model. The objective of MoI is to effect the transition from the real world to the virtual one in the MBSE approach. The method developed to support the MoI is based on system Interactions and Impacts (I&I). The I&I concept is to analyze a system by way of its complexity, which causes inter-system multidisciplinary interactions. The interface specifications from I&I supports the MoI building with port inputs, outputs and parameter information detection for a given scenario. Technically, the MoI is a model integrated in a multidisciplinary executable behavioral global model of the system under design. The objective of this global model is to show expected behavior of the artifact, which is represented by the MoI. The model's behavior includes performance requirements and dynamic tendencies, and guides the model providers and experts.

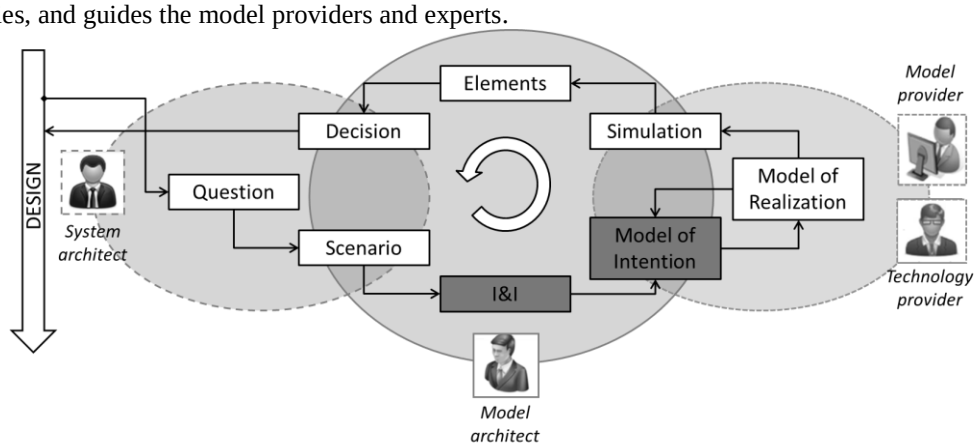


Fig. 6. Methodology to support simulation model building based on I&I and MoI – Dark boxes correspond to contributions

As shown in Fig. 6, from left to right, the method is explicitly decomposed into the real and virtual system specification and realization. The method is initialized by a design question that drives the model request. A first behavioral and executable model was developed in Modelica language that promotes energy

exchange in a powerful way [29]. To support propulsion system problems, we have developed a Modelica library to model different architectures easily. The methodology ends with the performance of simulations. According to the scenario, some results of these simulations can be selected to be sent to the system architect to support decisions.

### 3.3 PROPOSED METHODOLOGY: A MIXED APPROACH

The MIC and MoI concepts deal with the same objectives: reducing the knowledge gap between designer and model provider. However, they have complementary concepts for creating a complete model request package. The main difference is that MIC is a formal vocabulary and a MoI is an executable behavioral model.

The MIC tackles model specification with interface characterization issues, but its weak point is its low capacity for representing behaviors. These complementary characteristics allow a robust model request packaging such as (see Fig. 7 and Fig. 3):

- MIC for Structural simulation model design phase with interface specifications (S)
- MoI for Behavioral model design phase (B)

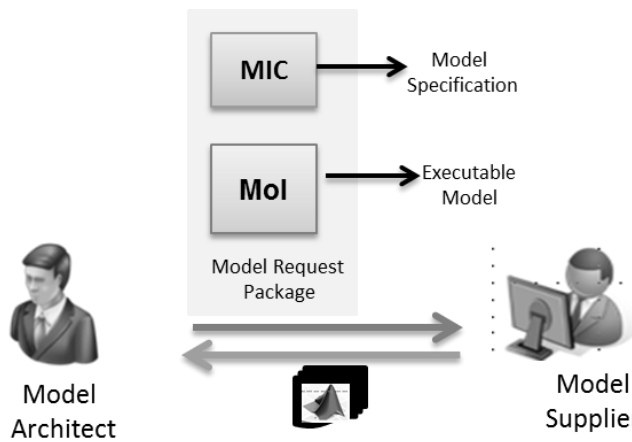


Fig. 7. Collaboration between Model Architect and Model Supplier: Information Flow 2

As we stated earlier, the challenge in using the suppliers' BB model is to take into consideration the number of distinct interface issues, parameters, or messages that have to be passed among the components. However, with our proposed clear model request package and formal model architecture design, integration of each BB model can be ensured.

As illustrated in Fig. 7, with such a model request package, expert and model supplier can manipulate the MoI observable parameters so as to be able to understand the requested models' expected behaviors for a given scenario. As illustrated in Fig. 8, the major role of the System Architect is to specify the system to be modeled and supply the essential system-related requirements to the Model Architect (see Fig. 3 Information Flow 1). By contrast, defined major roles of the Model Architect are designing a formal model architecture with a model-based systems engineering tool, specifying each sub-model with the Model Identity Card, and finally generating an executable MoI (see Fig. 3 Information Flow 2).

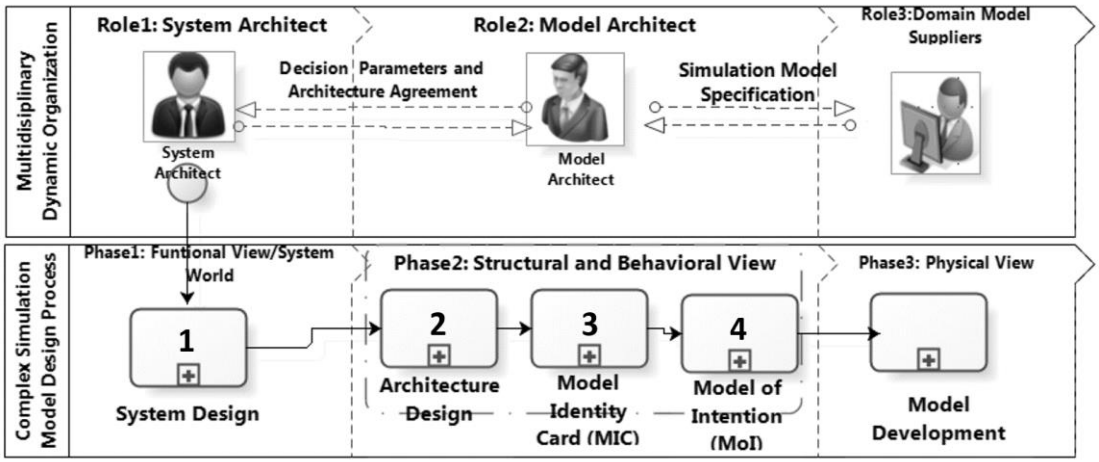


Fig. 8. Proposed Methodology

## 4. CASE STUDY

In this section, we illustrate the proposed method (see Fig. 8) with an industrial case study.

***First Step:* Inputs from the System Architect**

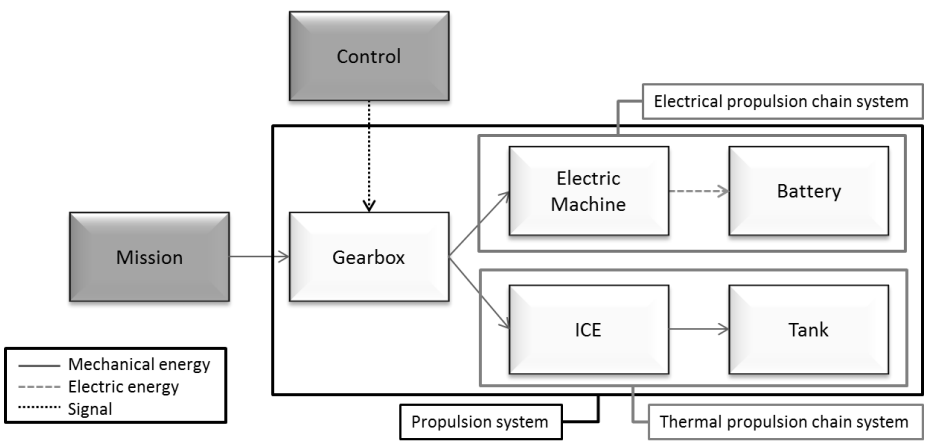


Fig.9. Hybrid parallel architecture

This case is built around a hybrid parallel propulsion system that contains five sub-systems (see Fig. 9). This architecture can be installed in various types of vehicle, such as a car or even an aircraft, and allows the product to having an additional power source when its mission requires it [21] [20], [22][21] and [29][28].
The scenario proposed by the system architect is to analyze and pre-size the new electric propulsion chain (technical objective) (see Section 2). Observables are the mass of this additional chain and the advantages in terms of pure power addition on a sizing objective mission (decision attribute). To obtain these observables, the system architect makes a requests to the model architect for a global model which that can simulate the

entire hybrid parallel propulsion system behavior, including models of the electric machine motor and battery. These two models must be as possible in line as possible with current technologies tendencies trends. The expected accuracy is to be around a 10% rate of error for each observable.
This section describes all the inputs that the system architect delivers to the model architect to complete his request (see Section 2). As visible shown in Fig. 9, the proposed architecture requires needs to be associated to with a mission objective and a control.
The mission objective is decomposed into five phases with different durations and required powers (P_requested).
The control is defined with a three-mode strategy based on power exchanges ($P_i$):

- Mode 1: $P_{requested} \leq P_{max\ ICE} \rightarrow P_{requested} = P_{ICE}$;
- Mode 2: $P_{requested} > P_{max\ ICE} \rightarrow P_{requested} = P_{ICE} + P_{Elec}$;
- Mode 3: $P_{requested} > P_{max\ ICE} + P_{elec}$; Power requested cannot be satisfied → mission fail.

The other inputs corresponding to existing and already specified components are the thermal propulsion chain and the gearbox models. This hypothesis is made in order to propose a reengineering problem, which reduces the design space and simplifies our demonstration.

***Formal Model Architecting***

The MoI uses the SysML modeling convention by default [36]. However, many engineers today are still unfamiliar with SysML, and the cost of training and licenses needed to put SysML tools into practice across large engineering teams can be prohibitive. In this paper we therefore adopt a different approach: rather than assuming that the structural model design is possible with a SysML environment, we prefer to use a system engineering tool named arKItect, because it enables us to specify the system architecture easily in a hierarchical manner. In this tool, the functional flows describe the interactions both between the system functions and between the system and the external environment. The flows can be either data or physical flows. Based on a powerful hierarchical type definition, arKItect lets us design our own metamodel very easily using a given metamodel structure: objects, flows and their composition rules (see Fig. 10) [9]. The design of formal system architecture is one of the activity areas of Model Architects (see Fig. 8) [36].
A part of the formal model architecture design is illustrated in Fig. 10. For clarity, we identify only one physical connection between Electric Motor and Battery sub-models.

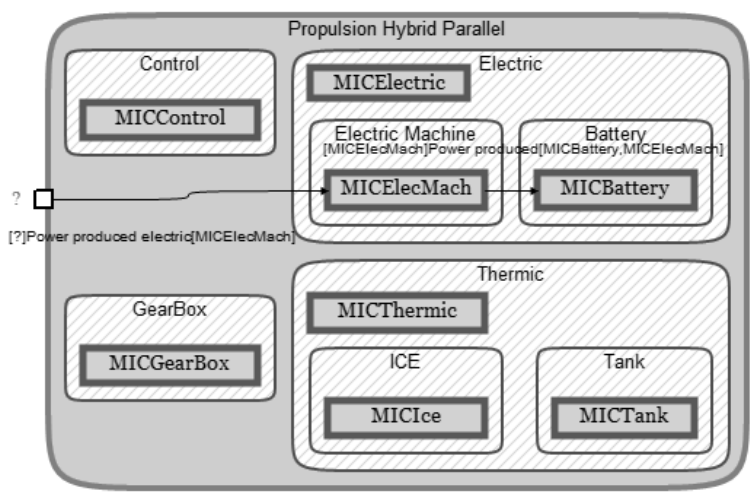


Fig. 10. Formal Model Architecture Design

Once the formal model architecture has been designed on the arKItect MBSE tool, we can identify each sub-model and system-level model with a Model Identity Card (MIC). Each sub-model has an MIC in which there is sufficient and necessary information about the related model and its port connection with other models.

***Second Step*: Model Identity Card (MIC) Creation**

Fig. 11 and Fig. 12 show the Electric Motor sub-model's specification with MIC. As stated in Section 3.1, the MIC graphical user interface contains two major parts: model characteristics and Port & Variables specification.

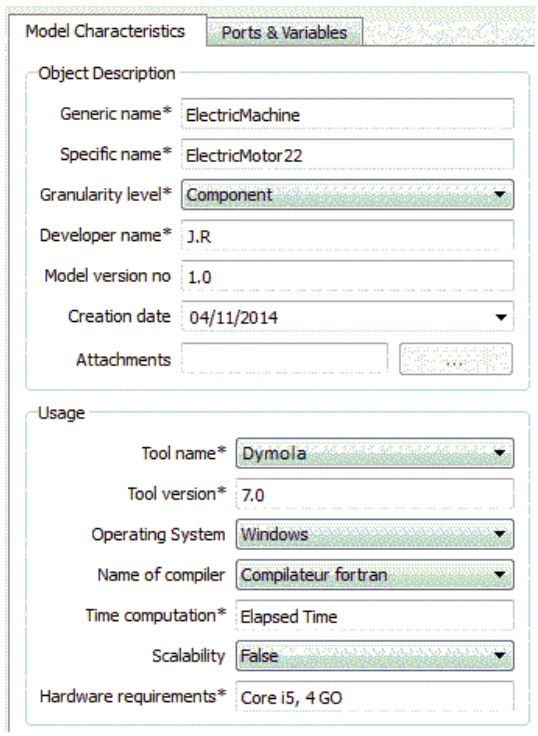


Fig. 11. Model Specification with MIC

The attributes we use in the MIC GUI reduce the ambiguity and misunderstanding between model architect and model provider. This possible communication problem has an adverse effect on model quality and decision. Some interoperability problems stemming from a misunderstanding of software versioning, undated libraries and model units, can be mitigated by using an MIC. The model provider also fills out some important attributes before pointing out the relevant model and sending the complete MIC to the model architect.

As shown in Fig. 11 and Fig. 12, the model provider must fill in important attributes, such as the name of the method that the model provider used to develop the model, and the model precision, solver name, dimension, time step, software tool name, versioning etc.

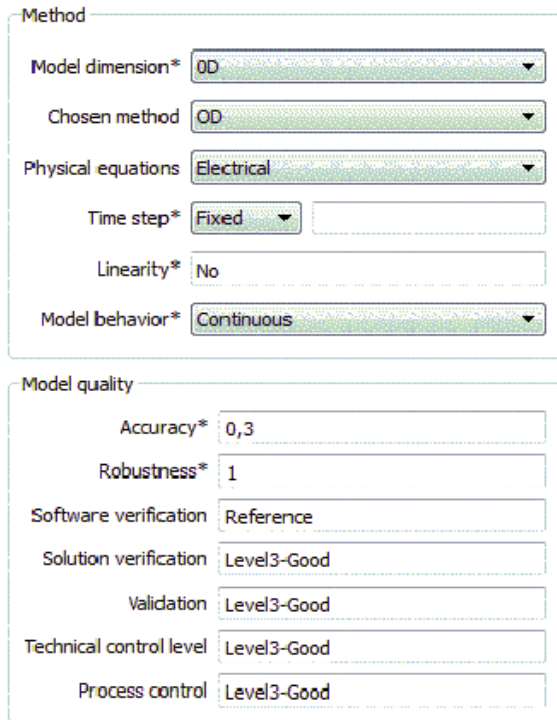


Fig. 12. Model Specification with MIC

As illustrated in Fig. 8, the model architect characterizes each incoming and outcoming port of the related sub-models and their connections with each other. Each port may have more than one variable, and we need to define the nature, direction, units, size and min/max value of each variable (see Fig. 7 and Fig. 8).

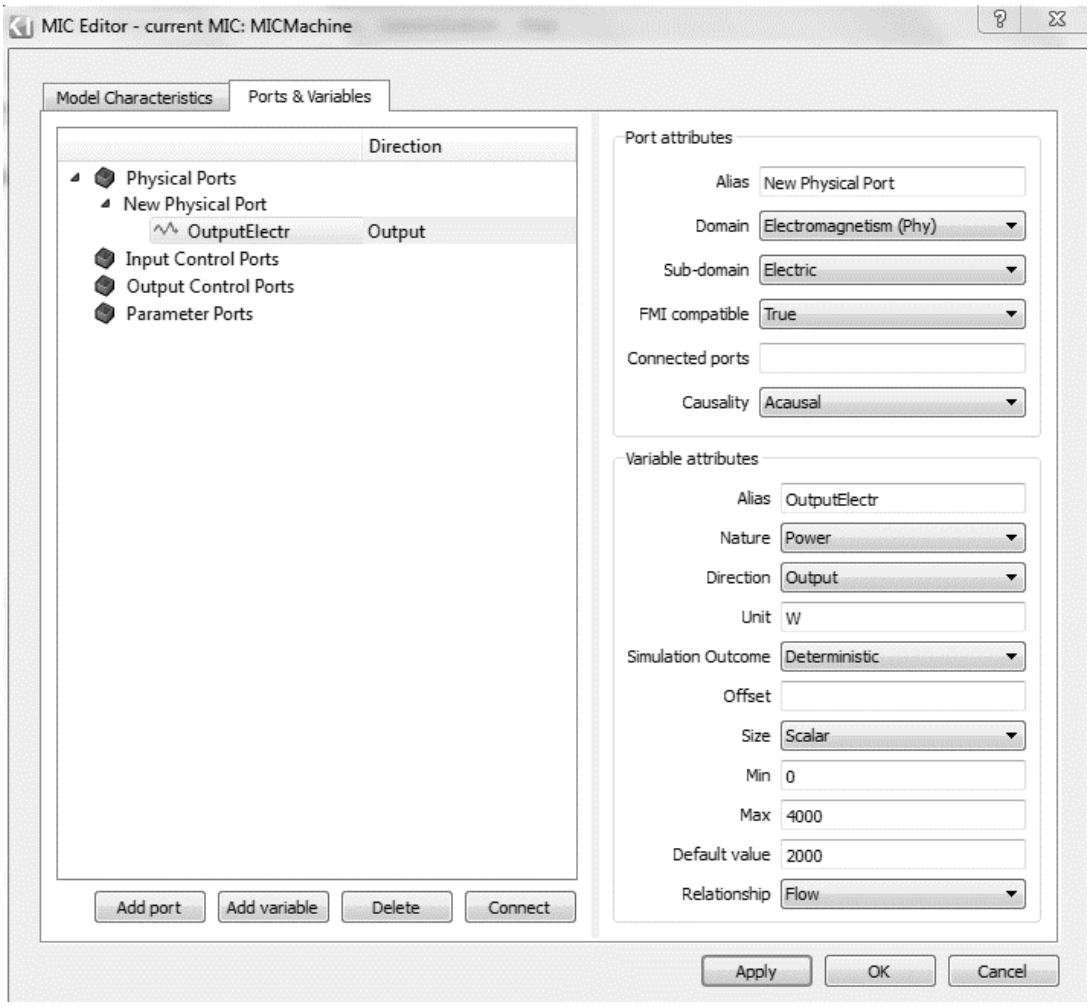


Fig. 13. Interface Specification with MIC

Finally, the MIC contains various correctness checks for interoperability problems such as domain model software names, versions, models' min/max values, units, the direction of a causal connections, models' accuracy levels, etc.

### *First Step : MoI Building*

In the MoI methodology, four types of model can be requested, on two axes: new model or modification of an existing one; system model or environment model. In this demonstration, a request for two new models is being made.

The first step in the MoI building is to propose an initial first model of architecture. Considering the gearbox and thermal chain already specified, models from the system architect are used. Control and mission models are customized with existing models of the specific propulsion system library that we have developed for MoI concept building.

All requirements, which can be modeled as inequalities in each model, are aggregated under a specific model to stop simulation if an inequality is false.

With the MIC specification, entire requested model interfaces are defined. In that way, it is possible to create models with ports and parameters directly in Modelica. These two new blank models (no equation inside) are introduced in the global architecture model (see Fig. 14)

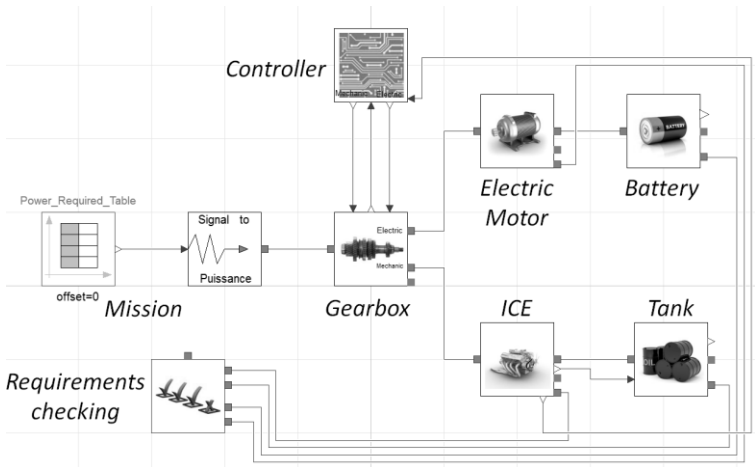


Fig. 14. Global model including electric motor and battery blank models

The next step in the MoI approach is to introduce simplified behavior that consists in connecting inputs and parameters to outputs, inside blank models.
For the electric motor (EM), two simple equations with two new factors proposed as parameters enable us to consider the two observables ($P_{\text{consumed}}$ and $P_{\text{max}}$):

- The electric motor's main function is to deliver rotational mechanical energy from electrical energy. In simplest modeling, the efficiency ($\eta_{\text{EM}}$) of this conversion is considered as a new parameter.

$$P_{\text{consumed}} = P_{\text{delivered}} / \eta_{\text{EM}} \quad (1)$$

- The electric motor mass is a parameter, as mentioned in MIC. A simple power density parameter ($\rho_P$) is proposed to relate mass and maximum power.

$$P_{\text{max}} = \rho_P . M_{\text{EM}} \quad (2)$$

These two behavioral equations are quite simple to introduce because they are based on the function or mandatory parameter, and are realistic because factor values can be specified with the help of the literature.
For the battery (B) it is only the energy transfer from battery to consumer (EM) that has to be modeled. This discharge is the main function because the loading function is not used, so not modeled:

- The battery main function is to deliver electrical energy from chemical energy. The efficiency ($\eta_{\text{B}}$) of this conversion is considered:

$$P_{\text{delivered}} = \eta_{\text{B}} . P_{\text{B}} / 100 \quad (3)$$

- The battery mass parameter enables us to determine the initial energy quantity, considering that the battery is fully charged at the start. To determine this energy, a simple energy density parameter ($\rho_e$) is proposed:

$$E_{(t=0)} = \rho_e . M_{\text{B}} \quad (4)$$

The real behavior is the energy consumption, which depends on the requested power. Based on the energy definition, we propose modeling the energy evolution with:

$$\frac{dE}{dt} = P_{\text{B}} \quad (5)$$

In order to precisely state our model request, a maximum power rate is introduced, with a time constant for discharge $(I_D)$ to calculate a max power:

$$P_{\max} = I_D . E_{(t=0)} \quad (6)$$

With Equation (6), the B model can be refined with the additional equation:

$$\frac{dE}{dt} = P_{\max} \text{ if } P_B > P_{\max} \quad (7)$$

Finally, if $E < 0$, no more energy is available, and the mission stops (i.e. the simulation stops).

The electric motor and battery MoIs are finished. All behaviors have been modeled with equations composed of factors that represent physical characteristics. The Values of these factors can be proposed as fixed, or with a max and/or min value in order to represent the design space given the supplier team (*cf.* **Erreur ! Source du renvoi introuvable.**). The final global model including electric motor and battery MoIs is visually similar to the model presented in Fig. 15, but with six new equations and three new parameters.

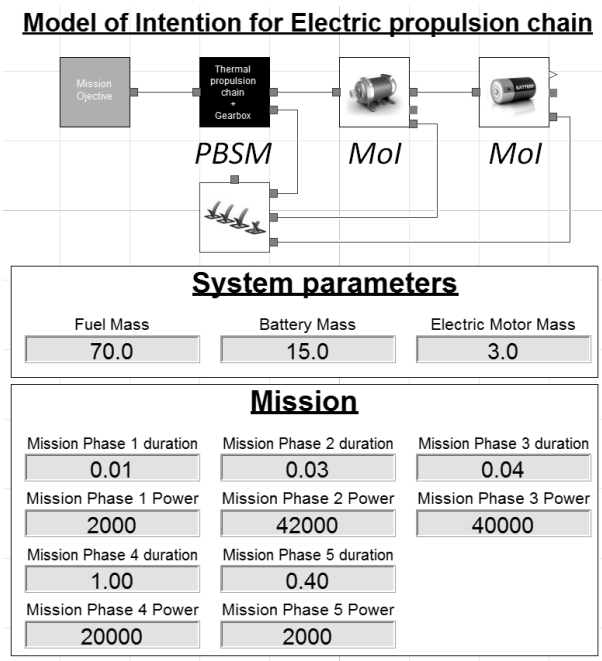


Fig. 15. Model delivered to suppliers

Table 1 Model parameters

| Model | Parameter | Value | Min | Max | Unit |
|---|---|---|---|---|---|
| PSBM | $M_{Fuel}$ | 70 | 1 | 100 | kg |
| EM | $\eta_{EM}$ | 95 | Unmodifiable | | % |
| | $M_{EM}$ | 3 | 1 | 40 | kg |
| | $\rho_P$ | 3000 | Unmodifiable | | W/kg |
| B | $\eta_B$ | 95 | Unmodifiable | | % |
| | $M_B$ | 15 | 1 | 40 | kg |
| | $\rho_e$ | 100 | Unmodifiable | | Wh/kg |
| | $I_D$ | 20 | Unmodifiable | | 1/h |
| Mission $i = [1,5]$ | $D_i$ | See Fig. 15 | Modifiable | | h |
| | $P_i$ | | | | W |

With the parameters in Table 1, numerous simulations can be performed. These induced results support the supplier team with additional behavioral information, which enables suppliers to hold discussions in order to propose advice on technologies to be introduced in their model of realization. For example, for the electric motor, the choice of a synchronous or asynchronous motor can be the subject of such advice. Simulation also allows the visualizing of system limits with requirement checking.

### 4.1 Comparative Analysis of Request Packages and Supplier Models

#### 4.1.1 Supplier model reception

After the reception of a model request package, each model supplier has developed a model called model of realization (MoR). The EM MoR is a synchronous permanent magnet motor model built around four equations and 11 parameters. All of these parameters concern the physical description of the motor (radius, pole number) or factors such as motor density or copper ratio. The battery MoR proposes a selection between two technologies (called Techno 1 and Techno 2).

Each technology has its own characteristic values inside its model. The global model, which integrates these two models in place of the respective MoIs, is the simulation model. When we check the Modelica global model, 92 equations are present for the global model with MoI, and 112 for global model with MoRs. Relatively, the difference of 20 equations shows the gain in complexity.

#### 4.1.2 Battery model comparison

The battery behavior needs to be studied alone owing to its dependency on the electric motor model outputs.

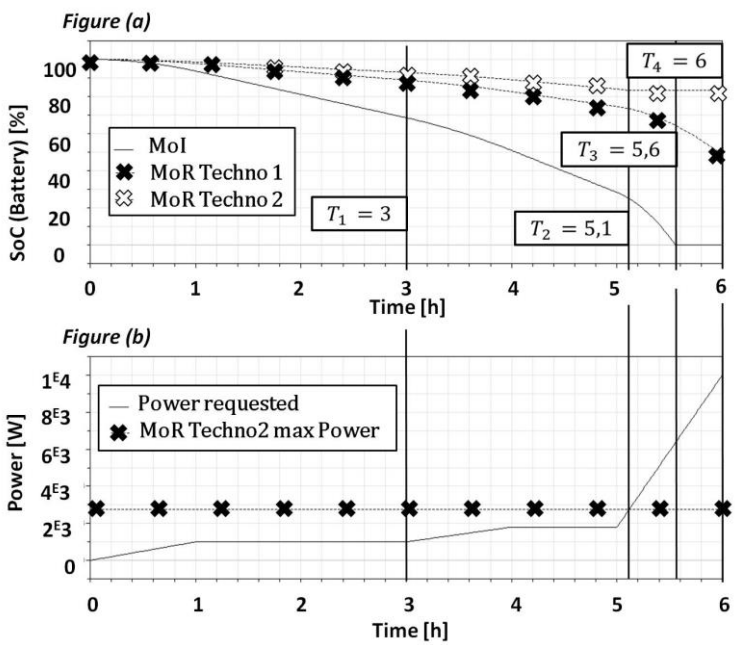


Fig. 16. Battery MoI - MoR comparison

Fig. 16 shows a local comparison between battery MoI and MoRs for battery masses equal to 80 kg. Fig. 16a is a battery discharge comparison between MoI and the two different technologies modeled with MoR. Fig. 16b shows in the full line curve the cycle used for this battery model test. The maximum power that technology 2 can deliver appears also to support the explanation for the top graph we propose in the next section.

Table 2 Battery model State of Charge (SoC) test

| $SoC$ (%) | MoI | MoR T1 | MoR T2 |
|---|---|---|---|
| $@T_1$ | 68.6 | 88.9 | 93 |
| $@T_2$ | 25.6 | 73.6 | 83.2 |
| $@T_3$ | 0 (empty) | 63 | out |
| $@T_4$ | out | 48.2 | out |

Table 2 groups the battery behavior at four simulation times:

- at $T_1$, a long iso-power request is finished. The different models show different SoC for the battery, the use of the MoI shows a deeper discharge than the MORs (Techno 1 and Techno 2);
- at $T_2$, power requested is too high for technology 2 because its power limit is 1100 W (MoR Techno 2 Max power curve is on 16.b). Technology 2 combined with its parameterization is unable to continue the mission;
- at $T_3$, the cycle begins a quick power increase phase. The comparison between MoI and technology 1 highlights a difference of more than 50%;
- at $T_4$, the mission is over. Only technology 1 has performed the mission.

The MoI is more pessimistic than the MoRs in the simulation results. In that way, model supplier and expert have understood the MoI because solutions are closer to technology and offer performances that are better than expected. However, each technology has advantages and drawbacks. Technology 2 is able to deliver more energy than technology 1, but technology 1 offers a faster power exchange.

#### 4.1.3 Global model comparison

After each global model simulation, the power requested from the electric motor is calculated at the controller component model. This power depends on the mission profile and the global model parameterization.

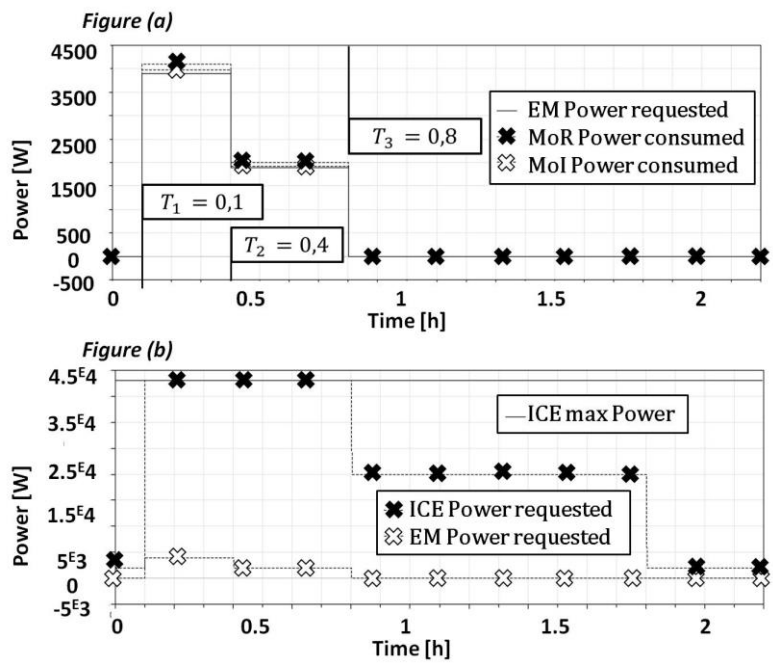

Fig. 17. Global model MoI - MoR comparison

Fig. 17 shows a simulation result for a mission. The first graph is the comparison of the MoI and MoR of electric motor power consumed, and the graph at the bottom is the power distribution between electric motor and ICE, of interesting especially when ICE cannot furnish sufficient power for propulsion (i.e. strategy mode 2). In curves ion the Fig. 17.a, we see that the electric motor is solicited between T_1 and T_3, with two different step values of which separation occurs at T_2. With the MoI, requested power is lower than with MoR, which is by hypothesis the more reliable model. However, the difference is low small (around 150 W). Technically, the MoR and its parameterization proposed by the supplier team is are close to the MoI: MoI modelling and simulation has supported the MoR building. In fine, the request package has successfully performed attained his its objective. From the system architect's point of view, the objective is to propose materials for the scenario. In that way, simulations performed by the model architect allow the identifying identification of how and when the electric motor is solicited. With curves in on the Fig. 17.b, the ICE is unable to furnish 3898 W between T_1 and⟦ T⟧_3, during for 0.7 h. In the scenario, the system architect wants to know the advantages of electrical propulsion chain addition before considering replacing or modifying the ICE. To estimate the architecture, mass balance is a material requested. In order to test the model, an optimization was done carried out with the minimization of a cost function composed of electric motor and battery masses and considering the mission success. A very first solution calculates an additional mass equals to 9.5 kg to the architecture (7.5 kg of Techno 1 battery and 2 kg for the electric motor). For a system architect, this kind of information supports his design decisions

## 5. CONCLUSION AND FUTURE CONSIDERATIONS

This paper addresses the common needs of software tool providers and industrial markets such as in the aerospace and automotive sectors. It also describes the methods that have been used to improve successful collaboration between industrial partners, and the challenging topics that still need to be explored in a multidisciplinary environment. Current inefficient ways of working create unnecessary iterations during the model integration phase, often responsible for increased product development lead time and cost, and schedule risk. To remedy this problem, the multidisciplinary models have to be understandable and easy to create. Thus it is necessary to maintain close links with the various disciplines and actors through clear information exchange.

Four major roles are identified: System Architect, Model Architect, Technology Provider and Domain Model Developer. This work also proposes a multi-view architectural method, aimed at reconciling the necessary high level and detailed design aspects with the help of the above actors. The integration of the two methodologies, MoI and MIC, narrows the knowledge gap between model designer and developer. The accuracy of the MIC, combined with the simulation capability of the MoI, represents a significant advantage. The freedom offered to the supplier is counterbalanced by the clear and direct model specification, in order to integrate it into a simulation chain. Misunderstandings are largely reduced, and a win-win collaboration is fostered. It is worthwhile considering an extension and refinement of the proposed method to support different specific domains of interest. The main long-term benefits of this work include significant reductions in time and in late inconsistency detection.

**REFERENCES**


[1] G. Sirin., C.J.J. Paredis, B. Yannou, E. Landel, E. Coatanea, "A Model Identity Card to Support Simulation Model Development Process in a Collaborative Multidisciplinary Design Environment," Systems Journal, IEEE, vol.PP, no.99, pp.1,12, 2014. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7004782&isnumber=4357939

[2] J.M. Branscomb, C.J.J. Paredis, J. Che, and M.J. Jennings, "Supporting Multidisciplinary Vehicle Analysis Using a Vehicle Reference Architecture Model in SysML". Presented at the Conference on Systems Engineering Research CSER, Atlanta, US-GA, March, 12-22, 2013.

[3] X. Zhang and J.F. Broenink, "A Model Management Approach for Co-simulation Model Evolution", In Proceedings of 1st Int. Conference on Simulation and Modeling Methodologies, Technologies and Applications, Noordwijkerhout, NL, pages 168-173, SciTecPress, 2011.

[4] G. Sirin, T. Welo, B. Yannou, E. Landel, "Value Creation in Collaborative Analysis Model Development Processes". Presented at the Conference on ASME (IDETC/CIE), Buffalo, USA, Aug., 17-20, 2014.

[5] B. Schätz, P. Braun, F. Huber, and A. Wispeintner, "Consistency In Model-Based Development," in Proceedings of ECBS 2003 10th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2003, from the department of Systems & Software Engineering at TUM (Broy).

[6] P. Andre, A. Romanczuk, and J. Royer, "Checking The Consistency Of UML Class Diagrams Using Larch Prover," in Rigorous Object-Oriented Methods 2000, York, UK, January 2000.

[7] Herzig, Sebastian & Qamar, Ahsan & Reichwein, Axel & Paredis, Christiaan. (2011). A Conceptual Framework for Consistency Management in Model-Based Systems Engineering. Proceedings of the ASME Design Engineering Technical Conference. 2. 1329-1339. 10.1115/DETC2011-47924.

[8] Simo-Pekka Leino, "Reframing the value of virtual prototyping", PhD Thesis, VTT Technical Research Centre of Finland, 2015.

[9] http://www.k-inside.com/web/fr/ressources/white-papers/

[10] P. Hester, (2012). Epictemic Uncertainty Analysis: An Approach Using Expert Judgment and Evidential Credibility, International Journal of Quality, Statistics, and Reliability, vol. 2012, Article ID 617481, 2012.

[11] X. Blanc, A. Mougenot, I. Mounier, and T. Mens, "Incremental Detection Of Model Inconsistencies Based On Model Operations," in CAiSE '09 Proceedings of the 21st International Conference on Advanced Information Systems

[12] Systems Engineering Fundamentals, DOD Supplementary text prepared by the defense acquisition university press fort Belvoir, Virginia 22060-5565, USA, 2001.
Available:http://123management.nl/0/070_methode/072_kwaliteit/Dod%20Systems%20Engineering.pdf

[13] Crescendo European Project. Participants' handbook. Crescendo Forum Innovation in collaborative modelling and simulation to deliver Behavioral Digital Aircraft 19-20 June 2012, France
Available:http://www.crescendo-fp7.eu/pages/downloads/crescendo-handbook-of-results.php

[14] S. Kleiner, C. Kramer, Model Based Design with Systems Engineering Based on RFLP Using V6, Smart Product Engineering, pp 93-102, DOI: 10.1007/978-3-642-30817-8_10, Springer 2013.

[15] C. Belton et al. (2003). A Vehicle Model Architecture for Vehicle System Control Design," SAE Technical Paper 2003-01-0092, 2003, doi: 10.4271/2003-01-0092.

[16] I.R. Grosse, J.M. Milton–Benoit and J.C. Wileden. (2005). Ontologies for supporting engineering analysis models. AI EDAM, Vol.19, No.1 p.1-18.

[17] F.N. Noy and McGuinness, (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford, CA.

[18] G. Mocko, R.J. Malak, C.J.J. Paredis, and R. Peak, "A Knowledge Repository for Behavioral Models in Engineering Design". 24th ASME Computers and Information in Engineering Conference, Salt Lake City, UT, Sept, 2004.

[19] Nasa Technical Standard, (2008). Standard for models and simulations National Aeronautics and Space Administration Washington, DC 20546-0001, NASA-STD-7009.

[20] D. Karnopp and R.C. Rosenberg. (1968). Analysis and Simulation of multiport Systems. The Bond Graph approach to physical systems dynamics, MIT Press.

[21] K.T. Chau, (2002). Overview of power management in hybrid electric vehicles. Energy Conversion and Management, 43(15), 1953-1968. doi:10.1016/S0196-8904(01)00148-0.

[22] C. Mi, M. Abdul Marsrur, D. Wenzhong Gao, (2011). Hybrid Electric Vehicles: Principles and Applications with Practical Perspectives, ISBN: 978-0-470-74773-5, 468 pages, June 2011.

[23] NASA, (2007). Systems Engineering Handbook

[24] INCOSE, (2006). Systems engineering handbook. INCOSE.

[25] C. De Tenorio, (2010) Methods for collaborative conceptual design of aircraft power architectures methods for collaborative conceptual design, PhD Thesis.

[26] P.M. Basset, A. Tremolet, F. Cuzieux, G. Reboul, M. Costes, D. Tristrant, D. Petot, "C.R.E.A.T.I.O.N. the Onera multi-level rotorcraft concepts evaluation tool: the foundations",
Available: http://addl.lsis.org/fr/manifs/jdl612/actes/articles/24.pdf

[27] S. Liscouët-Hanke, (2008). A model Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures. Architecture. PhD Thesis. INSA de Toulouse. Available: http://eprint.insa-toulouse.fr/archive/00000251/01/LiscouetHanke.pdf
[28] K. Amadori, (2008). A Framework for Knowledge Based Engineering and Design Optimization. Engineering. Available: http://www.diva-portal.org/smash/get/diva2:18279/FULLTEXT01.pdf.
[29] C.C. Chan, (2002). The state of the art of electric and hybrid vehicles. Proceedings of the IEEE, 90(2), 247-275. doi: 10.1109/5.989873.
[30] P. Fritzon, (2006). Principles of Object-oriented modeling and simulation with Modelica 21. IEEE press ISBN 0-471-47163-1
[31] RTCA and EUROCAE. Model based development and Verification, supplement of by RTCADo178c/Eurocae ED12C. Technical report, 2012.
[32] U.S. Dod. Verification, Validation and Accreditation Recommended Practice Guide (MSCO VV&A RPG) Glossary.
[33] IEEE Standard Glossary of Modeling and Simulation Terminology, IEEE Std 610.3-1989, vol., no., pp.0_1, 1989 doi: 10.1109/IEEESTD.1989.94599.
[34] F. Retho, H. Smaoui, J.C. Vannier, and P. Dessante, "Model of intention: A concept to support models building in a complex system design project", ERTS² 2014, France.
[35] F. Retho, H. Smaoui, J.C. Vannier and P. Dessante. (2014). Le modèle d'intention: Un concept clé entre architecte et expert pour la conception de systèmes complexes, Génie logiciel n°108, Mars 2014.
[36] www.uml-sysml.org.
[37] Pascal Roques, "MBSE with the ARCADIA Method and the Capella Tool", HAL Id: hal-01258014https://hal.archives-ouvertes.fr/hal-01258014 Submitted on 18 Jan 2016.