Simulation-based Multi-organization Engineering: Specification Application

Dr. R. Barbedienne, Dr. H. Sohier
(IRT-SystemX, France);

M. J. Silande
(ESI Group, France);

M. A. Levillain
(OPMobility, France);

M. C. Leclerc
(Renault Group, France);

M. M. Hayet
(Stellantis, France).

Abstract

Limiting carbon footprints is a global issue that has a huge impact on companies. Particularly in Europe, and in the automotive sector, where the sale of new combustion engine vehicles will be banned from 2035. These constraints require industries to integrate their new technologies into products as quickly as possible. This involves shortening the product development cycle. To reduce these development cycles, one solution is to offload development to suppliers. In this way, customer-supplier relations will become increasingly present.

This paper describes a tool-based process to help OEMs and suppliers to exchange simulation models. One of the challenges of this framework is that it must be able to adapt to companies with different vocabulary and operating modes. First, the difficulties were clarified in a workshop with experts from various OEM and supplier companies. This workshop showed that the main problems linked to model exchange between OEMs and suppliers come from the specification and credibility characterization phases of a simulation model. Existing solutions to solves these difficulties are a requirements list, defined by Nasa standard 7009B, the Predictive Capability Maturity Model (PCMM) or the Costa method. However, none of these methods is adapted to the most important criteria of the industry: the design phase, the expected maturity level of the model, and the expertise of the stakeholders.

This paper will present two solutions to address each of these issues. These solutions were co-constructed with several simulation experts from companies that are either OEMs or suppliers.

The first solution is a set of metadata (MIC core) to help with specification and a checklist composed of 24 requirements. Each requirement belongs to one of the five subsections: the clarity of the specification, the scope of the modeled system, the simulation environment, the model description, and the verification and validation procedure and criteria. These requirements are filtered according to three levels defined above, and filtered according to the MIC field that is filled.

The second solution is a credibility assessment questionnaire composed of 21 questions. Responses have been designed to be as interpretable as possible. With, for example, concrete thresholds to be reached or specific actions. These questions are used to calculate a score for 6 distinct categories: model robustness and sensitivity, model uncertainty and margin, expert verification, expert qualitative validation, experimental validation and model use. The main feature of this questionnaire is that some questions are completed by the supplier, and others are completed by the OEM to ensure that the model is used in accordance with the specification. This score is compared with a threshold, which depends on the 3 criteria mentioned above.

To assess the feasibility of implementing the approach, a demonstrator was created to support the approach. This article presents the application of the framework using a use case of integrating a fuel cell model into a thermal management model for an electric vehicle.

The proposed solutions address the issue of OEM-supplier interaction by improving both the specification process and the credibility assessment of simulation models. Future work will focus on the use of credibility assessment in simulation architectures composed of different models.

1. Introduction

a. Industrial context

Limiting carbon footprints is a global issue that has a huge impact on companies. Particularly in Europe, and in the automotive sector, where the sale of new combustion engine vehicles will be banned from 2035. These constraints require industries to integrate their new technologies into products as quickly as possible. This involves shortening the product development cycle. To reduce these development cycles, one solution is to offload development to suppliers. In this way, customer-supplier relations will become increasingly present.

For several years now, collaboration between original equipment manufacturers (OEMs) and suppliers has not been limited to the exchange of systems. Increasing complexity has forced OEMs to carry out more and more simulations. These simulations help to avoid issues when integrating sub-systems into the final system. As a result, suppliers have to exchange simulation models with OEMs. This model exchange is beneficial for the supplier, who can test the sub-system's behavior in the theoretical environment to which it will be subjected. Model exchange is also beneficial for the OEM, who can better understand the operation of the system delivered by the customer and adjust its system to improve the integration of the delivered subsystem.

b. Interaction between OEM and Supplier for simulation

The relationship between an OEM and a supplier is unique. The OEM has to find a compromise between the transparency of its specification and the risk of these elements being used by another of the supplier's customers. This compromise concerns principally component's environment. For its part, the supplier has to find a compromise between protecting its know-how and the trust that the OEM places in it. In other word, the customer-supplier relationship needs to encapsulate only what is necessary. In addition to these limitations, exchanges often face technological limits, such as exchange formats. Finally, the limitations that are often the most problematic are human ones, often due to misunderstanding or incomprehension between stakeholders.

In response to these problems, a number of technological solutions have been developed to improve exchanges. The FMI format [1] facilitate exchanges, by providing a compiled, black-box model to protect the supplier's know-how. It also improves compatibility between tools. However, the FMI format also has a number of limitations: it makes debugging more difficult for the OEM, and forces the OEM to contact the supplier again for any modification to the model, such as adding a new output or compile the FMU for another operating system.

To limit the risk of misunderstanding, companies are increasingly looking to formalize their simulation requirements and build a simulation architecture. As systems engineering made it possible to formalize technological choices in terms of systems, today simulation engineering could formalize choices in terms of modeling.

c. Simulation engineering

The concept of simulation engineering is not new. Since the 2010s, there has been a community trying to link the world of systems engineering with that of simulation [2], [3], [4], [5]. The goal of these ambitious approaches was avoid human error when creating a simulation. However, most of the approaches proposed were based on creating the simulation model, including the equations, in the system engineer's tool [6]. This meant that the simulation expert had to use a tool that was not suitable for simulation. The transition between model system and simulation can be made via a solicitation package, to avoid human error, as proposed by Sohier et al. [7] This promising approach allows OEM to specify a simulation model or architecture. This approach is complementary to the framework proposed in this document and focuses more on the link with system modeling.

Simulation tools are increasingly integrating the use of Simulation Process and Data Management (SPDM) [8] . This works well within a team or an organization, but the exchange between OEMs and suppliers is made difficult, as they don't have the same core business, and therefore not necessarily the same simulation tools.

d. Simulation engineering

To better understand the current constraints on customer-supplier relationships, a workshop was carried out with several simulation experts from Renault, Stellantis playing the role of OEM, and simulation experts from OPmobility playing the role of supplier. ESI Groupe was also present to represent the software world. This workshop showed that most of the difficulties between OEMs and suppliers concerning model exchange often stem from poor specification or documentation of the delivered model, and from a lack of confidence in the delivered model. This workshop showed that these difficulties can lead to loops between OEM and supplier, which lengthen design times.

The issue raised by this workshop and to which this article attempts to respond is How to ensure a continuity of simulation specification and simulation received between OEM and supplier?

2. Related Works

Simulation specification is not a topic much discussed in the scientific community. However, like system specification, good simulation specification is essential to avoid misunderstandings and delays in decision-making.

a. Conceptual model

The conceptual model [9] is defined as a means of describing a simulation model without the need for simulation tools. Liu et al[9] assert the following properties of the conceptual model:

- The conceptual modeling activity is iterative and repetitive throughout the development cycle.
- Conceptual modeling is a simplified representation of the real system.

- Conceptual modeling is independent of model or software code.
- User and developer perspectives are taken into account.

The conceptual model [10] is too close to the model and is not suited to the specification of a simulation model between an OEM and a supplier. Indeed, algorithms and assumptions must be developed by the supplier, who knows the system better than the OEM and is more likely to make the right assumptions. However, certain properties must be shared between the simulation specification and the conceptual model.

First of all, Pace [11] points out that the four qualities of a conceptual model are, completeness, consistency, coherence and accuracy.

Robinson [12] proposes a framework for building the conceptual model, the methodology of which is first to describe the problem situation, then to describe the general objectives of the problem, before defining the model output, then the inputs. At the end of this definition, the decision-maker must decide whether or not to use a simulation model. Finally, the contents of the model are described in this framework. This framework can be applied to an OEM-supplier relationship, if the first 3 steps are carried out by the OEM, the 4th by the OEM and the supplier, and the last by the supplier alone. However, the framework was not designed to deal with the problem of integrating the supplier's simulation model into the OEM's architecture.

Chwif [13] also propose a very interesting framework for specifying a discrete-event model based on a conceptual model. This framework consists of 4 parts;

1. Description of the objectives, complexity, inputs/outputs, and runs.
2. Process description, including model assumptions.
3. Detailed description of the input data.
4. Reviews and attachments table.

However, this framework poses two limitations for use in a customer-supplier relation context: firstly, it is specific to discrete-event model creation. Secondly, it doesn't take into account the simulation model's environment, so if the model is to be integrated into a simulation architecture

Finally, Fonces [14] demonstrates the need to validate a conceptual model before building the actual model. This validation also applies to the specification between OEM and supplier. Validating a specification avoids errors that could distort the supplier's construction of the model.

b. Others approach

The conceptual model is a good starting point for the proposed approach, but does not fully address the problem of bridging the gap between OEM and supplier. In their description of the conceptual model, Liu et al. [9] state that the conceptual model is built from simulation requirements. Stallinger and Grünbacker [15] addresses the concept of collaborative Requirement engineering. The proposed approach is based on the EasyWinWin requirement [16]. The aim of the approach is to collect, prioritize and negotiate requirements with all stakeholders. This approach is based on the principle of conducting several meetings with stakeholders, including researching the taxonomy of the domain concerned, Bratinstorming, categorizing, prioritizing, negotiating requirements, and finally mapping requirements to the taxonomy found. The approach emphasizes that these steps must be carried out in a positive, win-win situation. This approach is an excellent way of reaching agreement and avoiding unpleasant surprises when designing a simulation model. However, the approach has the drawback of taking a long time to perform. In a context where processors need to be faster and faster, this approach unfortunately seems too utopian.

Grotto et al. [17] propose an approach based on the Specification and Description Language (SDL) to formalize a digital twin in an urban mobility context. The approach has the advantage of taking into account all the elements of a model assembly, which has not yet been taken into account. However, the approach remains focused on system description and remains specific to urban mobility.

Gjerding et al [18] propose an approach based on the concept of Recipe and Record. Recipes are minimalist functions whose purpose is to execute code or actions. These recipes are based on data from records. The link with the specification is precisely these records, as they contain detailed information on the simulation, such as inputs, resources used and usage history. These records could be assembled to form a structured specification. However, the solution was not retained, as the report remained rather distant from the approach.

Finally, the Nasa std7009b standard [19] may at first appear to be far from the specification. This standard is a list of requirements to be validated throughout the model lifecycle. However, this is precisely the strength of this standard for specification. It ensures that the specification is built in such a way that the supplier meets these requirements. These requirements were analysed with the paper's co-authors, and they sometimes seemed not to be generic enough, and too close to the aeronautical context. What's more, in a context where processes between OEMs and suppliers need to be speeded up, these requirements sometimes seem too heavy to meet, particularly in a 0D/1D modelling context, during the pre-design phase. However, the concept of requirements has been retained to help specify the simulation.

3. Framework

a. Methodology

An overview of the framework used to facilitate the exchange of simulation models between OEMs and suppliers is shown in Figure 1: . This figure shows that exchanges between OEMs and suppliers take place mainly in 2 phases: the specification phase and the simulation model rendering phase. The workshop with simulation experts from Stellantis, Renault, OPmobility and ESI group, described in the introductory phase, showed that the main difficulties in customer-supplier interaction are due to these two phases.

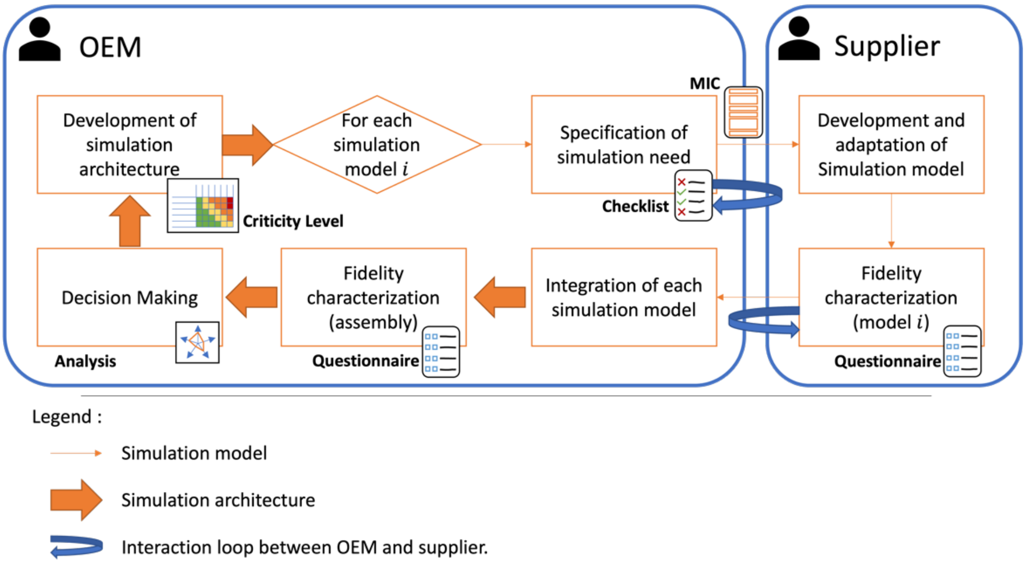

Figure 1: OEM supplier Cycle

During the specification phase, exchanges are mainly oral. Occasionally there are slides or email exchanges, but these are rare. Moreover, depending on the maturity of the system's development, the OEM is not always able to formulate his request correctly. This can lead to misunderstandings on the part of the supplier, who may deliver an unsuitable model to the OEM.

The second problem concerns model rendering. Here again, exchanges are often oral, with little documentation. There is more or less detail given by the supplier in the verification and validation phase, according to the expertise of the interacting stakeholders. Actually, giving too much detail to a novice engineer risks raising many questions about verification and validation choices.

For this reason, a framework has been built to help specify a simulation model, as well as to characterize the fidelity and credibility of a simulation model. As the framework is known to both the supplier and the OEM, the supplier can avoid questions about how to calculate model fidelity.

The specific tools built for this framework are the use of a MIC core to specify a model, and a checklist to help specify the model. The credibility of the simulation model can be assessed with a questionnaire.

The rest of this paper will focus mainly on the specification, but the next paragraph will give an overview of the questionnaire used to evaluate the fidelity of a simulation model.

The fidelity is assessed by means of a questionnaire. This questionnaire is composed of 15 questions. These questions are answered both by the supplier, to ensure that the model has been correctly developed, verified and validated, and by the OEM, to ensure that the model is used as specified. These questions fall into six categories: Robustness and model sensitivity; Model uncertainty and margin; Expert verification; Qualitative validation by expert; Experimental validation; use of the model. Note that the answers to the questions are used to establish a score for each category. This score is then compared with a threshold, which depends on the system modeling phase, the criticality of the model (which will be presented later) and whether the model will be integrated into a simulation architecture. Details will be provided in a further publication

b. Simulation specification dependency

The workshop presented in the introduction showed that specification depended on 3 major factors: the expected level of criticality of the model; the expertise of the stakeholders; and the system's design maturity.

i. Criticality assessment level

The expected level of model criticality is a major source of discussion between OEMs and suppliers. It enabling the supplier to know how much effort to devote to modeling, verification and validation of the simulation model. Most companies have a different scale for characterizing the expected level of model criticality. This can lead to disagreements between OEMs and suppliers, who don't necessarily have the same scales. There must be consistency between the criticality level expected of the model by the supplier and that specified by the OEM.

The model proposed here is based on the Nasa model (std 7009B) [19]. The expected criticality depends on 2 axes; the risk linked to the decision and the influence of the simulation model in decision making. The idea is that the simulation model will be used to make a decision. For the first axis, the more adverse the consequences of the decision (human risk, financial risk, etc.), the higher the criticality of the model. Similarly, for the influence of the simulation model on the decision, the more the simulation model is accompanied by other elements for decision-making, the lower the level of criticality.

Table 1: Criticality assessment

Simulation model results influence		I: Negligible	II: Minor	III: Moderate	IV: Significant	V: High	VI: Catastrophic
	5: Controlling	1	2	3	3	3	4
	4: Significant	1	2	2	3	3	4
	3: Moderate	1	1	2	2	3	3
	2: Minor	1	1	1	2	2	3
	1: Negligeable	1	1	1	1	2	2

In order to evaluate the decision consequence, the OEM must identify all of them for each category of the consequences of the decision (Personnel, Material damage, Project delay, project cost). The consequence of the final decision is equal to the category with the highest consequence.

Table 2: Evaluation of decision consequence

	I : Negligible	II: Minor	III: Moderate	IV: Significant	V: High	VI: Catastrophic
Personnel	No impact	Simple pain without physical alteration	Slight injury	Injury requiring external assistance for care	Major injury requiring rapid and costly care	Injury resulting in permanent disability or death of one or more humans
Material damage	No impact	repair costs < 5% of purchase price	repair costs < 15% of purchase price	repair costs < 50% of purchase price	repair costs >= purchase price	/
Project delay	No impact	delays with no consequences for customers	delays affecting pre-orders	delays that don't allow the company to take a leading position in the market, or that leave too much time for the competitors.	/	/
Project cost	No change	Margin forecast slightly affected (~10%)	Forecast margin reduced by half	forecast margin > 0	forecast margin ≤ 0 or project cancellation	/

ii. Expertise of the stakeholders

The experience of the stakeholders have an impact on understanding the specification. OEM must adapt the specification to the level of knowledge and expertise of the person who will be working on the project. It should be noted that when the expertise of the people interacting is not high, the equipment manufacturer must be particularly vigilant in his specification, clearly defining all the technical terms used and avoid technical jargon to avoid misunderstandings.

iii. System phase

The system phase has an importance concerning the precision of the specification. Indeed, it does not make sense to perform a 3D simulation analysis if the system architecture is not set. three design phases have been distinguished;

- **Predesign** or **conceptual design**: At this stage, the system architecture is not completely fixed. Simulation models are used to validate for basic conformity with requirements.
- **Detailed design**: The system architecture is fixed. Simulation models are used to ensure that the system operates as intended. Major modifications to the system architecture are expensive. The modifications concern the subsystems making part of the system architecture.
- **Digital prototyping**: This last phase is the most critical: it's the last simulations before the real prototype is made. In this phase, the cost of each modification can be very high. In this case, simulations are the most critical, as they often have to be as realistic as possible for a given scenario, while still being quick to run.

4. Simulation Specification

The three factors that are important for the specification have been defined, before considering how these factors will play a role in the proposed framework, this paragraph will focus on how the MIC core is used as a specification, and how it can be supplemented with a checklist to help specify a simulation model.

a. MIC core for specification

One of the challenges of specification is to succeed in formalizing the specification in a clear format that is easy for a supplier to understand. Specifications must also be comparable regardless of the software used to run the simulation. It allows for supplier to compare specifications and find the corresponding model.

MIC core is characterized by a list of metadata to characterize a simulation model throughout its lifecycle. MIC core can be extended with specific packages. In order to use the MIC core for specification, it can be complemented by the interfaces package, which contains the metadata for specifying the input and output ports, and therefore the parameters of the simulation model. The integration package can also be added when the simulation model need to be integrated into a simulation architecture. Integration package contains metadata describing usage software and hardware environment used to run the simulation model. This package helps avoid integration bugs. For example, if the model is complied to an FMU in the Windows operating system by the supplier, but OEM need to launch the model in a Linux operating system. There will be a bug when executing the model. Therefore the use of metadata allows to :

- structure the information;
- enable automatic interpretation of certain fields, such as ports or parameters, by a tool;
- make sure that OEM does not forget any important information.

More details on the MIC core fields are available on the MIC core github.

The MIC core is the first step in the specification process. However, the MIC core used on its own has the following limitations:

- it can not make sure OEM don't forget what's transverse to the MIC metadata, particularly with regard to the clarity of the sentences structure.
- it can not extend the scope of metadata, by specifying what information must be included in certain metadata depending on the context. In other word, the metadata of the MIC core are fixed, and do not adapt to the level of criticality of the model, the expertise of the stakeholders or the phase of the system.
- it can not request elements that cannot be entered as a Metadata (CAD files, real data, etc.)

b. Checklist

Therefore, the MIC core is completed with a checklist of requirements. This list of requirements can be adapted to the criticality level of the model, to the expertise of the stakeholders, and therefore to the phase of the system.

One of the main objectives of this framework is to be able to adapt to companies with different processes, different corporate cultures and different areas of expertise. Therefore, the requirements have been designed at the highest possible level, in order to adapt to the different companies and their different operating modes. A list of low-level requirements has been drawn up to group them together as far as possible to built high-level requirements. 24 high-level requirements was drawn up. These requirements were realized through interviews with business simulation experts playing the role of OEM or supplier. there are 5 subsections in which requirement are classified.

- **Clarity**: These requirements apply to all MIC core enriched with specification packages attributes. They are generic and describe the general clarity of information of the specification.

- **Scope of the modeled system**: The scope of the modeled system is the subset of requirements which refer to the description of the modeled system including parts, controls and system interfaces.
- **Simulation environment**: Simulation environment requirements describe all the outside of the simulation model. These requirements concern what is needed to run the simulation, but is often forgotten or neglected, such as the operating system, or the required libraries. These requirements are particularly important, because when they are not met, they are often the cause of many integration problems.
- **Model description**: The description of the simulation model concerns the requirements specific to the simulation model itself, such as inputs and outputs, or modeling choices. These requirements should not be overlooked by the OEM, to avoid numerous iterations with the supplier when he receives the model.
- **Verification and validation procedure and criteria**: Lastly, as the name suggests, the verification and validation procedure cover the special requirements of OEMs in terms of verification and validation. These requirements are important for a level 3 model or higher.

i. Filter and order

In order to filter the requirement, each requirement contains the threshold value for Design phase, Criticality level and Expertise of stakeholder. the example of requirement is given in the Table 3:

Table 3: Extract of requirements with threshold level for each category

Requirement	Design phase	Criticality level	Expertise of stakeholder
System footprint or geometry can be specified	Detailed design	3	High
The specification must describe the purpose of the mode	Predesign	1	Low

Thus, the requirements displayed are those whose threshold value in at least one factors (Design phase, Criticality level, and Expertise of stakeholder) is less than or equal to the factors assigned by the OEM.

For example, if only the 2 requirements in Table 3 are considered, and the specification is for a product in the detailed design phase, with a criticality level of 2 and a high level of stakeholder expertise, then both requirements will be displayed. In fact, for the design phase factors, all requirements have a threshold lower than or equal to the current phase (detailed design).

When filling in the MIC core, it is not necessary to display all requirements in the same order. Most requirements are linked to one or more fields in the MIC core. In the example in Table 3: , it is not relevant to display the requirement "The specification must describe the purpose of the model & Predesign" first, if the MIC field being filled in is the software and hardware requirement.

The idea is to order each requirement according to the distance between the selected MIC core field and the requirement. To calculate this distance, each requirement is associated with one or more MIC core attributes. As the MIC core has a tree structure, the distance is calculated by counting the number of edges before arriving at the nearest common node.

For example, in the Figure 2: the distance between selected MIC Core attribute and checklist requirement i is 1. the nearest common node is "Implementation", and there is only one edges before find this common node.

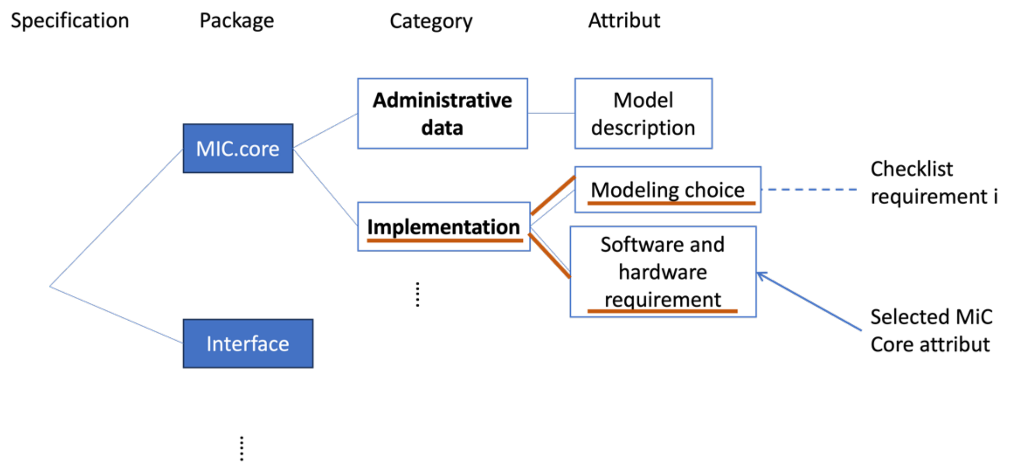

Figure 2: Estimation of distance between Mic core attribute and checklist requirement i

ii. Interface

The interface of the demonstrator is given Figure 3: . This interface contains the MIC core on the left and the checklist on the right. A toolbar allow to select the 3 factors described above. Note that these factors can be used to display or hide requirements.

Finally, this interface also shows that the order of requirements is adapted to the selected MIC core field.

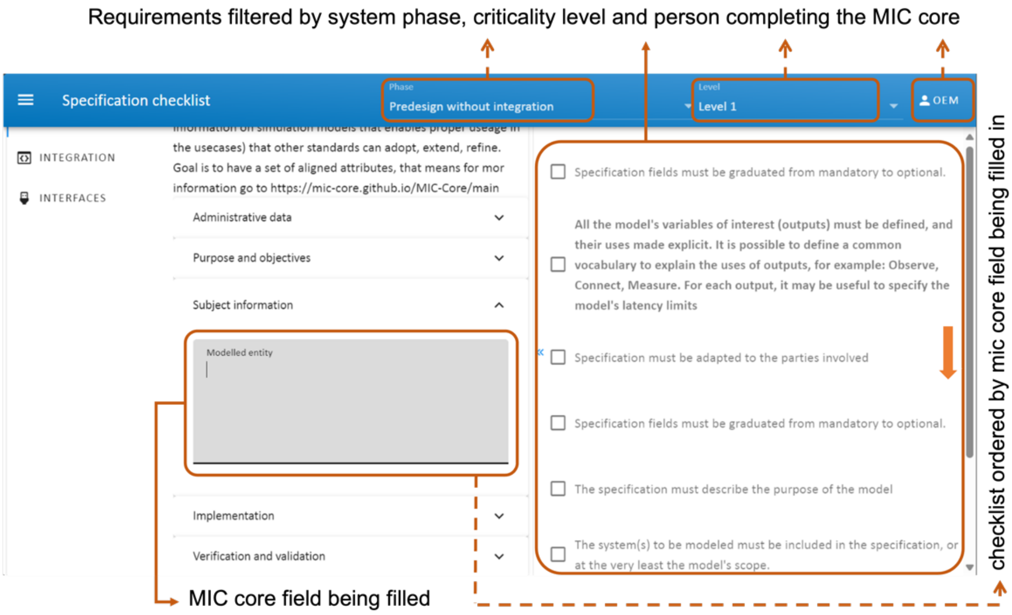

Figure 3: HMI of the demonstrator

5. Use case

a. Use case context and description

The main objective of this case study is to design a hydrogen-powered car. The OEM has chosen a supplier, in order to design a Hydrogen fuel tank. After iterations on the product, the OEM wants to evaluate the thermal management of the car. In a hydrogen vehicle, the hydrogen has to be heated in order to reach the fuel cell at the right temperature to maximize efficiency.

The OEM asks the supplier to provide a simulation model to represent the hydrogen tank. The simulation architecture is given in Figure 4: . For this first version, the aim is to ensure that the energy supplied by fuel cell is sufficient to power the thermal management and electronics. This means that the vehicle's thermal management system is not connected to the preheated hydrogen, only the power supply is connected. The model will be integrated in GT suite software. Gt suite is used as a master. The model to delivered by supplier is a tank and a fuel cell developed in AmeSim software.

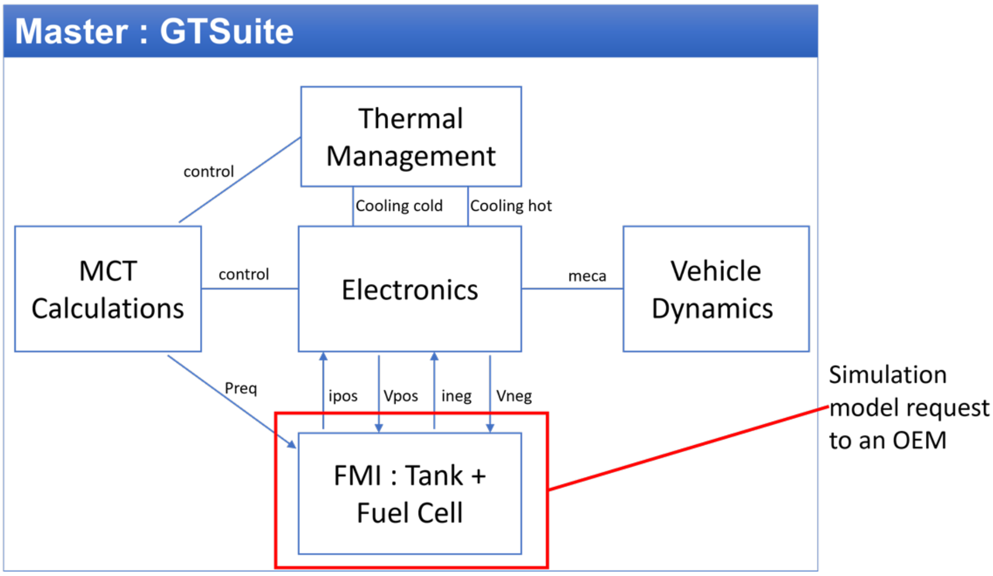

Figure 4: Simulation architecture

Before specifying the simulation model, the OEM must assess the simulation model's level of criticality. First, the influence of the simulation model is level 5, because, the decision will be taken only according to the results of this simulation. However, decision consequence is level 2 minor, because the design is still in the pre-design phase and a wrong decision would result in a slight delay. Thus, the criticality level of the model is 2.

Concerning other criteria, as it is the predesign phase, expertise of the interacting people is high.

b. Specification

Description of the demonstrator with MIC, and checklist

The specification can be performed using the MIC core enriched with integration and interfaces packages and the checklist. The Figure 5: is an abstract of the specification and the checklist. The checklist makes it easy to see when a requirement have not been met yet.

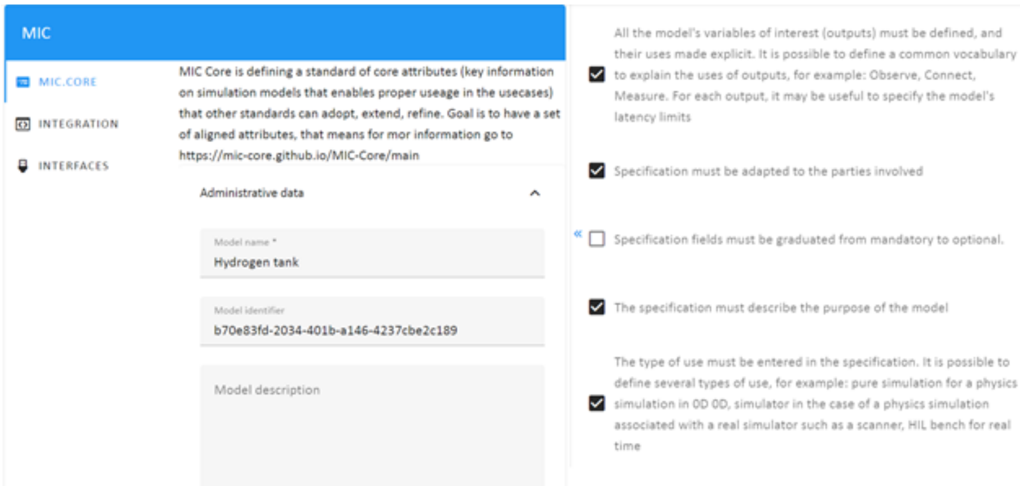

Figure 5: Hydrogen tank specification

The requirement which appears not satisfied in Figure 5: belongs to clarity category. It indicates that the fields must be graded so that the supplier can identify mandatory elements that may be an overlooked in the model. In this way, the OEM can rephrase these requirements to make the specification easier to understand.

c. Integration

The model were developed at OPmobility company. One of the difficulties of integration is to integrate a causal model into an acausal tool.

For this use case, the model was specified without using the checklist and the MIC. The specification process was carried out in the form of a meeting, with oral exchanges. First, the fuel cell system was described. This was followed by a presentation of the model and high-level assumptions. Finally, the interfaces between the models were discussed.

The model was delivered as an AmeSIM model. Usually, this model would have been delivered in FMI format, in order to keep the model in a black box. But for the purposes of this project, the model was transmitted in AmeSim format.

The integration of this model caused some integration problems. The integration problems encountered have been classified into two categories, those that could have been avoid by a better specification, and those that could not.

d. Problems that can be avoided using the proposed methodology

A number of problems were encountered in this use case, which could have been avoided if the specification had followed the proposed methodology.

i. Software version

The first issue met was that the software version. The model had been sent with a version of AmeSIM that was inferior to the OEM version. This problem was quickly solved by an AmeSim module that transformed the model into a version usable on AmeSim. However, this problem could have been more serious if the OEM version had been a lower version than the supplier's version. This problem could have been avoided by adding a constraint in the specification to deliver the model in FMU.

ii. Interface issues

The second problem occurred during integration. To work properly, the ECU needed to know how much hydrogen remained in the tank. This problem necessitated correcting the Amesim model to extract an output which was the tank pressure. The problem could have been addressed by the OEM at the specification stage. If the simulation architecture had been built before specifying the model, as specified in the checklist.

iii. Interfacing a causal model in an acausal master

One of the difficulties was the conversion between a causal and an acausal model. Correctors were added to the FMU model interface to connect the fuel cell model with an electrical circuit.

Although it is not recommended to impose on the supplier the tool to be used, the checklist does contain an indicator to explain in the OEM's specification to the supplier the use of the model. In the case of integration, it is precise to specify the nature of the tool to be integrated. Although in this case, the problem could not have been avoided, as the supplier could have provided us with a model on AmeSIM, the OEM could have prepared to receive a causal model and prepare the integration so as not to lose time during integration.

e. Integration problems difficult to anticipate

However, there were still a number of difficulties that were not easy to anticipate without a good understanding of the integration tool and the model that was delivered.

i. Parameter interpretation

For the first of these, when switching from the AmeSIM model to the FMU model, a model parameter was written using scientific notation. "X E-Y" with X a real number and Y a natural number. However, when the model was parsed, this number was interpreted by the FMU as a character string. The solution found was to change the unit of the parameter on AmeSIM to avoid switching to scientific notation.

This problem was difficult to anticipate, as it is specific to each tool's management of the parameter format. Another solution would be to have a database containing reports of integration incidents between different simulation tools, which would help to anticipate problems of connection between two specific tools, but would also run the risk of not being filled in by the person who carried out the integration.

ii. Variable outside Limit

Finally, a second limit which is also difficult to anticipate is that when the vehicle starts up, the power demanded by the vehicle's engine is very high. The power demanded by Gt suite as a result of the FMU exceeded the power definition interval. This caused the model to crash. The solution to this problem was to limit the power demand to the FMU limit by adding a saturation model. This avoids this error and is not a problem for the realism of the model, as a fuel cell would never have been able to deliver such a power to the vehicle.

Once again, this problem is difficult to anticipate. one solution would be to have a report of these incidents to anticipate the connection problem between two specific tools.

f. Use case conclusion

This use case shows that the proposed methodology avoids a certain number of integration problems, but there are still certain problem limits that will remain difficult to anticipate. These problems may be specific to the tools to be completed and even to the version of the tools that are coupled.

6. Conclusion and Future works

The framework propose a methodology to improve the OEM supplyer interaction. This paper addresses de specification step. The specification is perform using a set of metadata (MIC core) to help with specification and a checklist composed of 24 requirements. These requirement are adapted to the design phase, the expected maturity level of the model, and the expertise of the stakeholders. Approach was compared with a use case that had not benefited from this framework. This showed that certain integration problems could be avoided thanks to this framework.

The work presented here is a first version, and there is still scope for improvement. First of all, the proposed requirements can be improved. Indeed, some of these requirements deserve to be more generalized.

Today, checklist requirements have to be ticked off by hand, and there is no automated verification of requirements. A proposed improvement would be to automatically check these requirements using a Large Language Model (LLM).

Note that promising tests have already been carried out to pre-validate the feasibility of the concept.

Finally, one of the current limitations of the approach is the management of the simulation architecture. Today, the specification is based on a simple model without taking into account the simulation architecture. This is important in order to avoid forgetting the interoperability. It also means that integration constraints, which are the same for all models in the simulation architecture, are not rewritten.

7. Acknowledgment

Many thanks are addressed to the project partners and simulation experts who took the time to answer our questions, and without whom this framework would not exist.

8. References

[1] "Functional Mock-up Interface Specification." Accessed: Jan. 31, 2025. [Online]. Available: https://fmi-standard.org/docs/3.0/

[2] C. J. J. Paredis *et al.*, "5.5.1 An Overview of the SysML-Modelica Transformation Specification," *INCOSE Int. Symp.*, vol. 20, no. 1, pp. 709–722, Jul. 2010, doi: 10.1002/j.2334-5837.2010.tb01099.x.

[3] W. Schamai, P. Fritzson, C. Paredis, and A. Pop, "Towards unified system modeling and simulation with ModelicaML: modeling of executable behavior using graphical notations," in *Proceedings of the 7th International Modelica Conference; Como; Italy; 20-22 September 2009*, Linköping University Electronic Press, 2009, pp. 612–621.

[4] R. Barbedienne, O. Penas, J.-Y. Choley, and P. Hehenberger, "Modeling Framework for a Consistent Integration of Geometry Knowledge During Conceptual Design," *J. Comput. Inf. Sci. Eng.*, vol. 19, no. 021009, Mar. 2019, doi: 10.1115/1.4042551.

[5] G.-D. Kapos, A. Tsadimas, C. Kotronis, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "A Declarative Approach for Transforming SysML Models to Executable Simulation Models," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 6, pp. 3330–3345, Jun. 2021, doi: 10.1109/TSMC.2019.2922153.

[6] T. Johnson, A. Kerzhner, C. J. J. Paredis, and R. Burkhart, "Integrating Models and Simulations of Continuous Dynamics Into SysML," *J. Comput. Inf. Sci. Eng.*, vol. 12, no. 1, p. 011002, Mar. 2012, doi: 10.1115/1.4005452.

[7] H. Sohier, P. Lamothe, S. Guermazi, M. Yagoubi, P. Menegazzi, and A. Maddaloni, “Improving simulation specification with MBSE for better simulation validation and reuse,” *Syst. Eng.*, vol. 24, no. 6, pp. 425–438, 2021, doi: 10.1002/sys.21594.

[8] Nafems, *What is simulation data management?*, Nafems Ltd. in Na. Accessed: Jan. 31, 2025. [Online]. Available: https://www.nafems.org/downloads/sdmwg/nafems_wt02_-_what_is_simulation_data_management.pdf

[9] J. Liu, Y. Yu, L. Zhang, and C. Nie, “An Overview of Conceptual Model for Simulation and Its Validation,” *Procedia Eng.*, vol. 24, pp. 152–158, 2011, doi: 10.1016/j.proeng.2011.11.2618.

[10] S. Robinson, R. Brooks, K. Kotiadis, and D.-J. van der Zee, Eds., *Conceptual modeling for discrete-event simulation*. Boca Raton: CRC Press, 2011.

[11] D. K. Pace, “Ideas About Simulation Conceptual Model Development”.

[12] S. Robinson, “Conceptual modelling for simulation Part II: a framework for conceptual modelling,” *J. Oper. Res. Soc.*, vol. 59, no. 3, pp. 291–304, Mar. 2008, doi: 10.1057/palgrave.jors.2602369.

[13] L. Chwif, J. Banks, J. P. De Moura Filho, and B. Santini, “A framework for specifying a discrete-event simulation conceptual model,” *J. Simul.*, vol. 7, no. 1, pp. 50–60, Feb. 2013, doi: 10.1057/jos.2012.18.

[14] P. Fonseca I Casas, “A Continuous Process for Validation, Verification, and Accreditation of Simulation Models,” *Mathematics*, vol. 11, no. 4, p. 845, Feb. 2023, doi: 10.3390/math11040845.

[15] F. Stallinger and P. Grünbacher, “System Dynamics Modeling and Simulation of Collaborative Requirements Engineering”.

[16] A. M. Christie and M. J. Staley, “Organizational and social simulation of a software requirements development process,” *Softw. Process Improv. Pract.*, vol. 5, no. 2–3, pp. 103–110, 2000.

[17] A. Grotto, P. F. I. Casas, A. Zubaryeva, and W. Sparber, “Formalizing Sustainable Urban Mobility Management: An Innovative Approach with Digital Twin and Integrated Modeling,” *Logistics*, vol. 8, no. 4, p. 117, Nov. 2024, doi: 10.3390/logistics8040117.

[18] M. Gjerding *et al.*, “Atomic Simulation Recipes: A Python framework and library for automated workflows,” *Comput. Mater. Sci.*, vol. 199, p. 110731, Nov. 2021, doi: 10.1016/j.commatsci.2021.110731.

[19] *Standard for models and simulaion*, NASA STD 7009B, 2024. [Online]. Available: https://standards.nasa.gov/sites/default/files/standards/NASA/B/1/NASA-STD-7009B-Final-3-5-2024.pdf