



HAL
open science

Value creation in collaborative analysis model development processes

Göknur Sirin, Torgeir Welo, Bernard Yannou, Eric Landel

► To cite this version:

Göknur Sirin, Torgeir Welo, Bernard Yannou, Eric Landel. Value creation in collaborative analysis model development processes. IDETC 2014: International Design Engineering Technical Conferences / CIE: Computers and Information in Engineering, Aug 2014, Buffalo, United States. pp.DETC2014-34696, 10.1115/DETC2014-34696 . hal-01786644

HAL Id: hal-01786644

<https://hal.science/hal-01786644v1>

Submitted on 28 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

DETC2014-34696

VALUE CREATION IN COLLABORATIVE ANALYSIS MODEL DEVELOPMENT PROCESSES

Göknur Sirin^{1,3}
goknur.sirin@ecp.fr

Torgeir Welo²
torgeir.welo@ntnu.no

Bernard Yannou¹
bernard.yannou@ecp.fr

Eric Landel³
eric.landel@renault.com

¹ Ecole Centrale Paris, Laboratoire Génie Industriel, Chatenay-Malabry, France

² NTNU, Engineering Design and Materials, Rich. Trondheim, Norway

³ Renault SAS Techno-Centre, Guyancourt, France

ABSTRACT

Integration and coordination of engineering analysis model is a vast development field in the context of complex product development. Engineers' siloed way of working in combination with lack of efficiency in current model development process may cause inconsistency based on model interfaces, human errors, miscommunication between teams and misinterpretations. In lean terms, this may create multiple wastes, including waiting, over-production leading to excess inventory, unnecessary processing and may be the most harmful: defects (e.g., incorrect models) with rework consequences. Hence, product manufacturing companies must establish effective processes to add value throughout the multidisciplinary distributed modeling environment. The goal of this paper is to propose a pull-control model development process, providing model architecture integration and coherent control in early design phase. This paper proposes also an appropriate reuse strategy; this allows for utilizing plug-and-play type modular product models managed through a single-source of authority concept. A pull-control development process helps prevent potential rework arising from inconsistencies related to definitions, know-how and stakeholders communication at an early stage of the design process. Also, the proposed black box models reuse strategy helps reduce human-related error such as lack of domain knowledge, experience and

misinterpretations. The proposed method is used to identify and visualize potential improvement in terms of increased model transparency and reuse when transforming from the present to the suggested future modeling strategy. The research has been conducted by synthesizing findings from a literature review, in combination with observations and analysis of current analysis model development practices within the automotive OEM Renault in France.

Keywords— Modeling and Simulation, Lean Systems Engineering, Collaborative Product Development, Model Reuse, Pull and Push Concept

1 INTRODUCTION

A. Background

Nowadays analysis models are becoming critical tools for automotive vehicle development. These models can be used to better understand the impact of different operating conditions on vehicle attributes and performances. To accurately represent vehicle behavior in a given field situation, these analysis models are created by integrating different models of the subsystems within a vehicle, such as chassis, engine, transmission, etc. These subsystem models are referred to as domain models. However, vehicle modeling is a complex endeavor which includes multiple disciplines, coordination between multiple stakeholders (i.e. Internal and external model suppliers, model architect and system architect), management of large amounts of data and information throughout the lifecycle of the modeling process [1]. Today, one of the major challenges is to manage the creation of system level models by integrating associated domain level model and providers, while delivering a viable system level model in less time. Since the quality of the complex multidisciplinary system model is strongly dependent on input from multiple sources, domain experts from different disciplines must establish a common understanding for collaborating and transforming knowledge across the different disciplines [1].

Hence, ensuring confidence in simulation results (i.e., model credibility) is particularly challenging for the design of multidisciplinary systems in which domain models within different disciplines are tightly coupled. In addition there are many factors that may cause inconsistencies, such as model interfaces, human error, miscommunication between teams and misunderstood assumptions. These inconsistencies are all sources to uncertainty and its propagation in multidisciplinary modeling environment is more complicated than in a single disciplinary domain. The effect of the uncertainties in one domain model may propagate to another through interrelated variables, and the final system output will suffer from the accumulated effect of the individual uncertainties. Thus, the information flow in modeling practice is one of the key aspects of its uncertainty, which may imply risk that the product attributes will not ultimately meet user needs [2]

In a vehicle development environment, the modeling process involves several companies (suppliers) and several design teams evolving through large-scale partnership [3]. A large amount of data is then created by the different partners (company internal and external suppliers) involved in designing a new product using communication and IT technologies to access and exchange product design information in a context of simultaneous and concurrent systems engineering [3]. Consistency between the models in the design process within a socio-technical system is not only governed by the technology tools available but also to a large degree by human factors, such as confidence in model quality as well as company cultural factors such as procedures for appropriate model reuse and more

formal communication. On the contrary, for example, lack of a well-defined model reuse strategy can be a source for inaccuracy, leading to uncertainty and time delays as data is lost or is not consistently modified in the translation from one stakeholder to another. Here the term model reuse refers to modifying existing domain models to fulfill purposes for which they were not originally made. Model reuse is an important overarching objective since it can reduce development time and decrease costs. However, model reuse requires well defined documentation along with towering engineering knowledge and experience about the system [4]. On the other hand, if the engineers lack critical parts of domain knowledge, model validation can be as time consuming and costly as adequate models have to be developed from scratch [4]. The viewpoint defended in this paper is that unless a Modeling and Simulation (M&S) practitioner understands the model's contextual dependencies accurately and unambiguously, model reuse will continue to be an ineffective trial-and-error effort [18]. To this end, the basic motivation for building an adequate system level model is to encourage using domain models as a black box. To be able to integrate the domain level models to a full-vehicle system model, we need to prepare a detailed virtual prototype of the system in the early design phase. This need requires integrating multiple views (functional, structural and physical) and leveraging a flow-correctness check mechanism to the current model development procedure. It helps also to synchronize between actual needs of the downstream process and what the upstream process is delivering (see Fig.1 right side image). Today, however, most of the high level model development and integration activities do not integrate this kind of early detailed model design and correctness check. In this manner, the system level model integration activity is conducted by finding, modifying and integrating existing models (physical prototype) without knowing which sub models are meaningful to connect together and, more specifically, how they can be connected to each other and with which interface requirements (see Fig.1 left side image). It is typically a Push way of working.

B. Objectives and Research Questions

The solution proposed to overcome some of the problems indicated above is a detailed Model Design phase integration and a correctness check in early model building process (see Fig.1 right side image). Thus, definition of project's scope with different views (i.e., functional, structural and physical) in the design phase vastly influences the model development and its overall performance. Understanding the complexity of design in functional, structural and physical contexts at an early stage is important in defining appropriate end facility of the project [5].

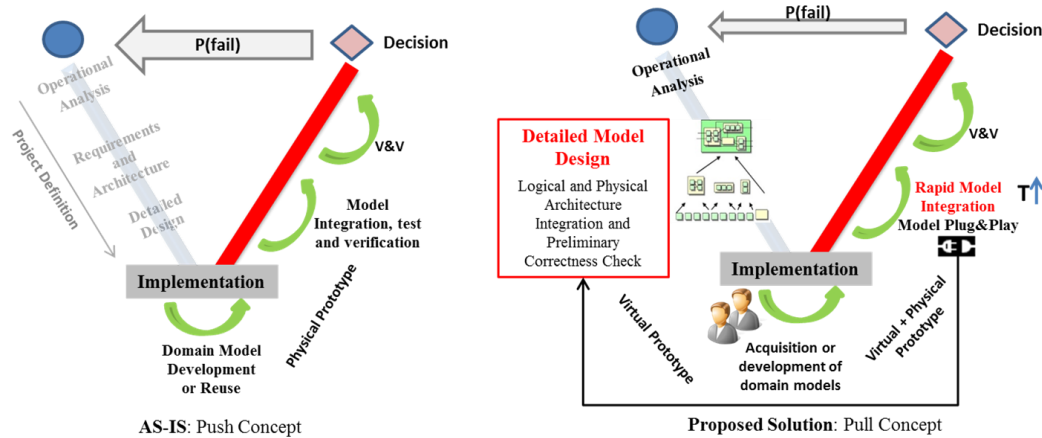


Figure 1. Push and Pull Concepts

Today, many companies try to use the V-cycle system engineering process for product development as proposed by Frosberg and adapted by the National Council on Systems Engineering [6]. However, companies' model building process is typically functioning as a push system rather than a pull system (see Fig.1 left side image); i.e., a push system is one where an upstream operation transmits work to a subsequent downstream operation without being requested as a need for further processing. The push concept, also called 'over-the-wall-design/modeling', is only efficient when looked at from a local, silo-structure's view point. In a push-type model development process, testing and integration of hardware products are mainly done in later process stages using physical prototypes. This may create multiple wastes, including waiting (i.e., the model is not available when needed or it is sitting waiting for somebody to process it further), over-production (i.e., the model is not needed), which may lead to excess inventory (i.e., data/models not utilized), unnecessary processing (i.e., sending files/models not requested or recreating existing models), and may be the most harmful of them all, namely, defects (i.e., incorrect models), which is subsequently used as basis for design decisions. The common denominator is lack of understanding of needs of intermediate 'customers', causing rework [7].

An alternative model development process must be more effective and efficient than the existing. The aim of this work is to maximize the value creation throughout engineering analysis model process, while increasing the confidence in the model. In this connection, a suitable measure may be to apply principles from Lean Thinking to the Systems Engineering (SE), combining these two concepts into a common modeling strategy, henceforth called Lean Systems Engineering (LeanSE) [8-9]. Hence, LeanSE may be a suitable strategy for achieving improved modeling practices in systems engineering. Adding a Detailed Model Design stage and Correctness Check in the current model development process improves the traditional V-cycle because this allows problems to be identified early. As shown on the right hand side of Fig.1, this may reduce rework in the more expensive

implementation and physical prototype validation phase, which is the main driver for product development cost [10].

To maximize the value in collaborative modeling environment depends on advances in many areas. Among the key issues here are:

- (a) Creating a pull-type model building process (Preliminary Model Design / Virtual Prototype) based on the actual needs of immediate model users;
- (b) Defining a model sharing and reuse strategy as an enabler to reduce model recreation and errors associated with new model construction and;

Addressing these two issues will help identify wastes, inefficiencies, and non-value added activities, which should be eliminated come up with at a more desirable "future state" as an intermediate stage towards the possible "ideal state" serving as a longer-term goal.

The remainder of this paper is organized as follows. In Section 2, we provide an investigation and literature review about Complexity management and Value creation. In section 3, we introduce our methodology to optimize the current model development activities by proposing a Pull Process with a detailed Model Design Phase and Correctness Control and a Model Reuse Strategy. The conclusion and future research are in Section 4.

2. DETAILED INVESTIGATION AND LITERATURE REVIEW

2.1 Managing Complexity with Different Views and View Points

The applicability of any model depends on the accuracy and reliability of its output. Yet, 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 value and true value. High accuracy implies a low error. Precision is the closeness of the agreement under the same conditions, and reflects the repeatability of the result [12]. Moreover, input data errors and modelling uncertainties are not independent of each other as they can interact in various ways.

Sources of uncertainty may be categorized in model inputs, numerical approximations, or in the form of mathematical models. Input uncertainty arises when the requirements that define a design problem are inaccurate. Model inputs include both parameters used in the model of the system and data from the surroundings (see Fig.3). Model input data includes parameters such as geometry, constitutive model, and boundary conditions, and can come from a range of sources including experimental measurement, theory, other supporting simulations, or even expert opinion [2]. Data from the surrounding includes boundary conditions and additional knowledge (see Fig.3). Additionally, in a design process there are different stakeholders such as model and system architects, model providers, who have different needs and viewpoints. In Model Based System Engineering, Views and Viewpoints can be used to model the perspectives of different stakeholders and their interests. A viewpoint describes a particular perspective of interest to a set of stakeholders, while a view is a stereotyped package that is said to conform to a particular viewpoint [13]. In a traditional document-based approach, each stakeholder works from their own domain-specific tools and documents that they need to perform their tasks. For a model-based approach, this necessity remains. Ideally, each stakeholder would only use models which have been customized to their “view” of the system, in lieu of any documents. Practically speaking, however, this is not yet possible. Documents and presentations are still an integral part of the engineering design process. To support this functionality for a SysML-driven MBSE approach, many tools have been developed that allow users to generate and modify documents linked to a SysML model [11].

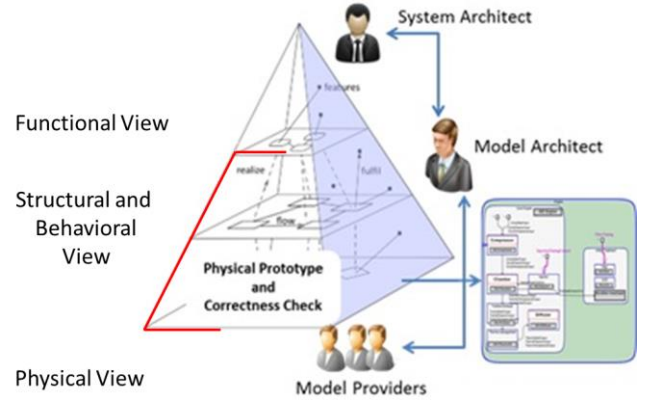


Figure 2. Views and View Points

As shown in Fig.2, the proposed Detailed Model Design phase contains three views: Functional view identifies the boundaries of the system, the requirements and defines what the system has to accomplish for the users. Structural and behavioral views illustrate the system as a white box and define how the system will work to fulfill expectations. Physical view indicates how the system will be developed and built, along with interfaces and domain model specifications with model providers. These three views provide the basis for creating a virtual prototype in the early design phase. After establishing these three views, the next step is to send a request to domain model suppliers. The request has to contain the industrial criteria and integration strategy, including what is acceptable for each supplier. The process that the authors explained in the section 3 covers especially Structural and Behavioral views creation.

2.2 Lean Thinking and Value Creation

In Lean SE, value is defined simply as mission assurance (the delivery of a flawless complex system, with flawless technical performance during the product or mission life cycle), satisfying the customer and all other stakeholders, which implies completion with minimal waste, minimal cost, and the shortest possible schedule [14].

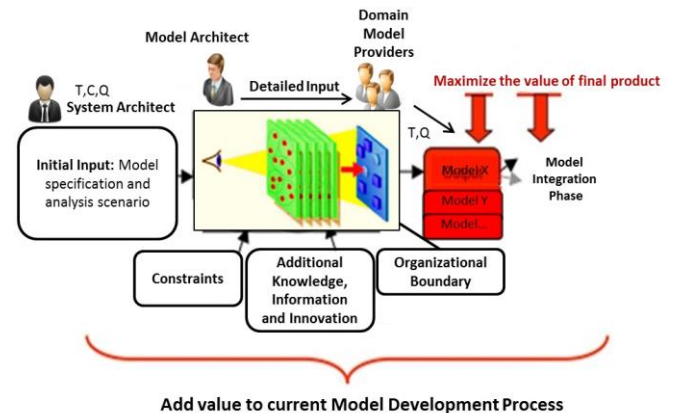


Figure 3. Value Creation in Modeling Process

As illustrated in Fig.3, the aim of this work is to maximize the value of the analysis model. This means to increase the confidence in the analysis model by decreasing time to development and cost. To be able to evaluate added value, addition has to be paid to the current Model Development Process. However, identifying value in the model development process is not straightforward. The complexity of the process, distance from the final customer, shifting market conditions, and uncertainties of technical performance, cost, and schedule, all make a simple definition of value based on customer (model user) needs unworkable for process improvement [15].

Overall, creating value through Product Development Process (PDP) activities includes creating information and knowledge about the product definition while reducing risk and uncertainties in the project. The activities require internal and external inputs supported by resources (human, material, IT resources) to provide added-value outputs.

In this paper, without going into details in Lean philosophy, we would like to inspire from lean thinking to be achieved through the identification, monitoring, analysis and continuous improvement of value chains in the Modeling and Simulation activities. The purpose is to create a continuous flow of material and information to deliver the desired customer value with the least possible waste of resources and minimized delays. In "Lean Thinking" [14], the authors define an iterative approach, inspired by the Deming's "quality wheel", sequenced in 5 basic steps:

1. *Defining what makes or creates value for customers*
2. *Identifying the value stream*
3. *Promote the flow of the stream by ensuring that the stages of value creation are optimized*
4. *Pull the flow downstream*
5. *Striving for perfection to achieve excellence*

We will try to improve the current model development process based on these 5 basic steps.

2.3 Non-Value Added Activities in Product Development Process

Today, supplying analysis models especially from external providers may be challenging. For example, when supplying a model from the requirement elicitation phase to model integration tests, the probability of failing is very high since, among others, there is limited common vocabulary between the two stakeholders. The source of the problem is also based on wrong or insufficient knowledge transmission from automotive manufacturer to model suppliers. Due to lack of common understanding and transparency, the result may be that the provided model does not fully conform to the requirements (Fig.4). As a final result the mentioned activities take a lot of time, sometimes as much as 1-6 months, of several problem solving meetings and integration tests.

Based on our observation in Renault, the final product performance is mostly related to the following three key points:

- **Knowledge generation and appropriate model reuse**

The knowledge encapsulated in each analysis model must be standard and coherent to be used by another party. The

knowledge must be captured for reuse in future projects. To reduce the possible risk caused by inappropriate model reuse, therefore, one needs to define a robust model reuse strategy. Here documentation alone is insufficient; equally important is model providers' towering knowledge and experience. The model reuse strategy is discussed in Section 3.

- **Common understanding and Team communication:**

Knowledge sharing is one of the key points in the model development process. Providing a common vocabulary for the M&S users can help communicate fact-based decision in a maker's assessment of the credibility of M&S results. The ability for users to select from a list of options is an immensely important capability. Because creating full-vehicle simulation models is a multidisciplinary process, it is important that the same strategies are used across different teams of domain experts. By limiting large groups of users to the same vocabulary and set of options whenever possible, inconsistencies arising from miscommunication or misinformation can be reduced significantly.

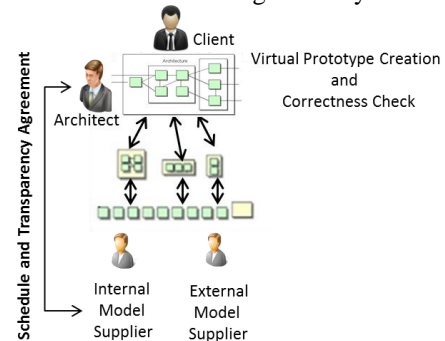


Figure 4. Principal actors and communication in complex model development strategy

- **Trust in provided model quality:**

Among inhibitors to consistency management, humans who create or provide models can be a primary cause when not conforming to the common language, or laws of nature, or introducing changes in a model. To this end, model provider or developer develop fit-for-purpose instructions model.

A simulation model and its results have to be credible for the decision-makers to accept them as "correct." Note here that a credible model is not necessarily valid. The following items may increase credibility of a model:

- The decision-maker's (model user) and model suppliers' understanding and agreement with the model's assumptions,
- Demonstration that the model has been verified and validated,
- Reputation of the model developer (provider),
- Importance of detailed model design and correctness check in early development phase.

As shown in Table1, the authors define 5 categories of waste in PDP adapted by Oppenheim: Waiting, Inventory, Over production, Defects and Transportation. The examples and causes are developed based on our observations in the

Table 1. Waste in PDP adapted by Oppenheim [16]

Type of Waste	Examples	Causes
Waiting Unavailable information, Late delivery of information, <u>Lack of resource</u> , <u>Scheduling</u>	<ul style="list-style-type: none"> • People waiting for information • Participant Availability 	<ul style="list-style-type: none"> • Lack of access and poorly designed or executed process to provide information • How much time will each participant be able to provide the modeling effort? • What time frames are available for inter-team coordination?
Inventory Complicated retrieval; Outdated, Obsolete information	<ul style="list-style-type: none"> • Too much information • Multiple/redundant sources • Outdated/obsolete information 	<ul style="list-style-type: none"> • Poor understanding of user needs • Tendency for everybody to maintain Lack of “version control” • Lack of disciplined system for updating new and purging old information • Inadequate archiving standards or practices
Over Production Creation of unnecessary data and information, Information over-dissemination	<ul style="list-style-type: none"> • Pushing, not pulling model • Lack of reuse strategy 	<ul style="list-style-type: none"> • <u>Poor understanding of each participant’s needs</u> • Send all information to everyone, rather than to meet specific needs • Tendency to “over-design” • More detail than necessary in early design
Defects Erroneous model information and reports. Lack of reviews, tests, verifications and validation, Lack of interpretation and responsibility	<ul style="list-style-type: none"> • Errors in data reporting/entries • Errors in information provided to customers • Information does not make sense to user 	<ul style="list-style-type: none"> • <u>Human error, lack of expertise</u> • Poorly designed input templates • Lack of disciplined reviews, tests, verification and validation • <u>Lack of model specification and early correctness control</u>
Transportation Unnecessary movement of information between people, organizations, or systems, Information and Software incompatibility	<ul style="list-style-type: none"> • Delay • Refactoring 	<ul style="list-style-type: none"> • Lack of clear information flow paths, failure of process to produce information needed • Incompatible information types (drawings vs. digital descriptions) • Incompatible software systems or tools • Lack of availability, knowledge, or training in conversion and linking systems • Software/hardware incompatibilities

Company. Since these 5 categories are strongly linked to the project targets [22].

Process overview:

However, we can argue that, due to the collaboration process in product development, the delivery delays in products have considerable financial impacts for all collaborators in this process. Some studies have shown also that delays in development were factors that had the most negative impact on expected profits, and that exceeding the development budget was the factor with the least impact [14-15]. In addition, managing several projects simultaneously is not a trivial issue, especially

for companies developing different complex products like automobile engine. One of the key issues is the resource allocation and finding the balance between single project optimum and overall organizational benefits.

In addition to this observation, we would like also to highlight that poor understanding of client’s needs, human error and lack of expertise are the also the key issues in complex model development process. This is the reason why, section 2 and 3 introduce Pull-type model building process with a Detailed Model Design stage and Early Correctness Check and the most reliable Model Reuse Strategy.

3 METHODOLOGY

3.1 PULL-TYPE MODEL BUILDING PROCESS

This section aims to demonstrate the flow and pull concept for downstream activities (see Figure1 right side). The “Flow” principle enables the value creation process to flow smoothly and continuously without waste, such as unintended stops, waiting, rework, or backflow. On the other hand; “Pull” promotes the culture of tailoring tasks and their outputs to meet the legitimate needs of internal or external customer, while eliminating wasteful activities. In addition, the flow principle contains several measures related to the practices intended to boost the flow. These include frequent clarification of requirements as well as frequent opportunities for decision-making, using effective communications and coordination practices [7-8]. The major aim of this detailed design phase is to promote the communication and engineering data exchange between different design actors especially between external model providers and model architect(s).

Actors:

The auteurs propose a high-level overview of Detailed Model Design process as illustrated in Figure5. Note that there are three actors shown here: **System and Model Architect** and **Model Providers**.

System Architects are the sponsor of model development activity. He or she defines the projects’ expected time, cost and decision parameters issues. **Domain Model Providers** are the domain experts who build models with theirs specific domain knowledge. Model Design activity contains formal architecture design with domain models’ interfaces definitions, vehicle level and domain level model specifications including an early interfaces consistency control between specified interfaces. Model Design phase gives a structural and semi-behavioral view about the system to be modeled. Thus, this transversal view from Functional to Physical View should be managed by a new actor of the collaboration named “**Model Architect**”. The authors would like to highlight that, in today’s design environment, there is no or not clearly defined role of the Model Architect. In this proposed detailed model design process, we suppose that each Model Architect has a multidisciplinary vision of a product,

and simulation knowledge. They have also a deep understanding of both the system-level requirements for the vehicle model, as well as how their models must interface with other domain models.

Activities Overview:

Each activity addresses different questions. Par example; analysis application plan of System Architect addresses; detailed description of set of analyses to be run such as what will the analysis be used for, which vehicle architecture are we considering? Etc...Model Requirements contain vehicle and domain model requirements. In vehicle model requirements, we try to know which simulation environment will be used, which interfaces are needed etc. In domain model requirements activity; our aim is to know what assumptions should be made, are there assumptions or restrictions that will affect other domains etc.

holistic view of a system and allow different stakeholders to work together with a common basis in the same vehicle system definition [17]. There are many different architecture exist for capturing vehicle architectures; for example, if the goal of a system design study is to examine different vehicle architectures, such as a traditional internal combustion engine version or a full-hybrid version. It is of the utmost importance that this information is presented explicitly; otherwise it could result in major inconsistencies between the models create by different domain engineers. In particular, communication may be challenging between model architect and domain model suppliers since they in most cases are not collocated. Defining lists of model attributes for model specification and most vital attributes to reduce interoperability problems is very important. Domain engineers must specify both the control

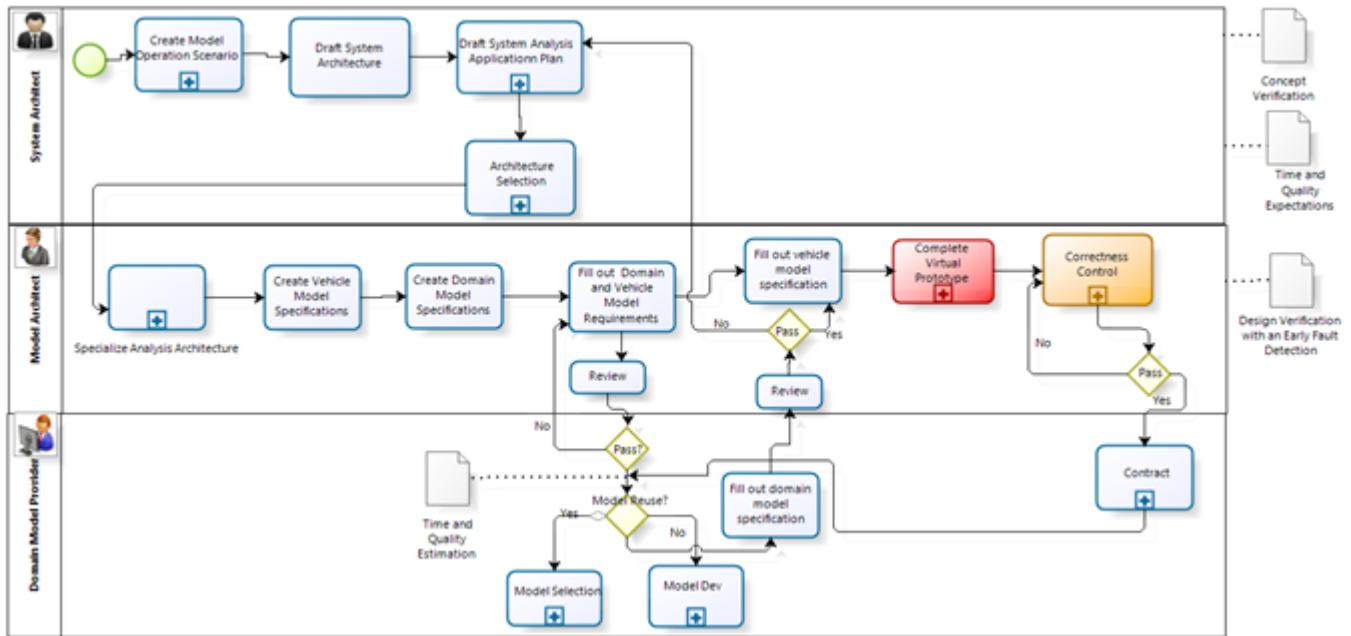


Figure 5. Proposed Pull Type Model Design Process

Domain and Vehicle model specifications addresses the questions such as how should the model be created, how should it be verified and validated? How should the domain models be integrated and how should the integrated models be verified and validated? Based on output from the previous steps, the System and Model Architect can draft a set of requirements for the vehicle system model. This includes specifying the computer and operating system that the model should be run on, which solvers and file formats to use, which language(s) the model should be written in, etc. These requirements are used to drive the development of the individual domain subsystem models, whose requirements are defined at a later stage. System architecture selection and characterization is extremely useful in complex, multidisciplinary vehicle system analysis. Architectures provide a

attributes that they need and the attributes that they plan to provide from their models. This complete list of model attributes is then reviewed by model architect, who negotiate with each domain team to develop a consistent set of signals for the entire vehicle. These system and model architects must negotiate with both the domain engineers providing signals and those receiving signals, so that all of the analysis models are compatible. Because this is an iterative process, it may require several rounds of negotiations with all of the different teams before a common vehicle-wide set of control signals can be agreed upon. Using a formal check list and a correctness control is critical for early virtual prototype validation.

Correctness Control

After having established the domain level signals (interfaces and domain level model specifications), the model architects integrate these sub models into a full virtual prototype. The model architect can evaluate alternative architectures against different model accuracy constraints. S/he has to detect any potential problem before the IVVQ (Integration, Verification, Validation, and Qualification) phase. The virtual prototype contains various correctness checks for interoperability problems such as domain models software names, versions, models' min/max values, units, the direction of acausal connections, models' accuracy levels, etc. (see Figures 6 and 7) [12].

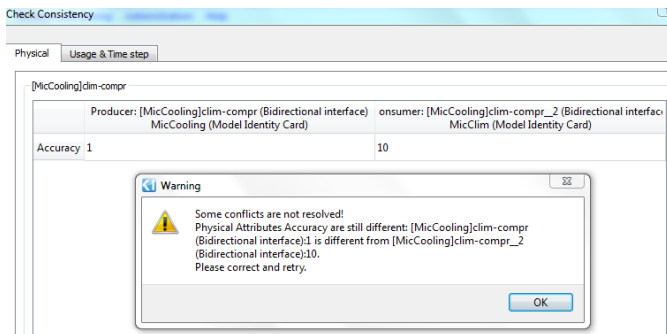


Figure 6 Accuracy control between ports

	Compilability	Time computat	Scalability	Software	Software versic	Time step
MicThermot	Yes	Elapsed tin	No	Amesim		hour
MicCooling	Yes	Real time	Yes	Amesim	V12	hour
MicAirext	Yes	Elapsed tin	No	Amesim		hour
MicClim	Yes	Elapsed tin	No	Amesim		hour
MicHvac	Yes	Elapsed tin	No	Amesim		hour
MicHabitable	Yes	Elapsed tin	No	Amesim		hour
MicClimbox	Yes	Elapsed tin	No	Simulink		hour

Figure 7 Time Step, Software and the other controls

In previous sections, we introduced a Detailed Model Design and Correctness Check of complex model development. The next sub section is developed for the aims at clarifying the viewpoint defended in this paper about model reuse strategy.

3.1 Modeling and Simulation Interoperability and Reuse Strategy

This section presents model reuse strategies (refactoring, reuse and plug-in) which provides advantages and disadvantages in achieving balance between model credibility and development time and cost. By reusing a model, designers avoid the expensive and time-consuming task of developing a new model. Designers often adapt models published in reference library, and they copy computer code to make up parts (or all) of their new model (a practice known in the computer engineering community as “code

scavenging”), and they invoke software components they have written or purchased previously. However, multidisciplinary analysis model reuse is much more risky than software code reuse because the former requires engineer’s deep domain knowledge and system experience [4].

Reuse has different meaning for different modelers. In one case, reuse could be limited to only recomposing existing models from a library, without modification of components. In other contexts, reuse can involve both reuse without modification, as well as modifying an existing component if it meets modeling needs and/or reuse will speed development.

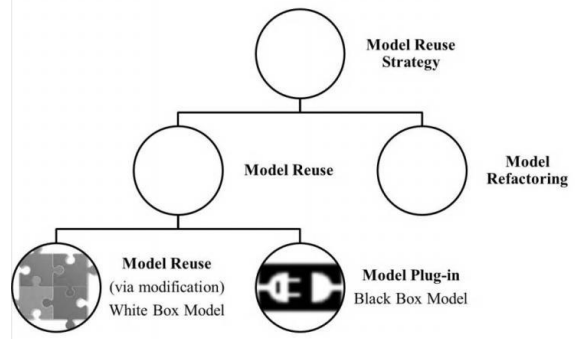


Figure 8 Model Reuse Strategy, General View

Model Refactoring

The way that engineering teams exchange modeling and simulation data is often siloed and highly inefficient. One of the primary consequences of silo structure is model refactoring. The term refactoring refers to developing an existing model from scratch. Change control and version management tend to be manual or more or less absent, and models and simulation data are often stored on local drives or network shares. Different engineering teams in the same organization may be solving the same problem, or even one that has already been solved, and they lack effective ways to achieve good solutions. Refactoring is the source of wastes like inventory, over-processing and over-production. As a result, companies try to find alternative solutions to refactoring for saving time and money. One of the alternatives to model refactoring is establishing a modeling reuse strategy. If correctly developed, reusing existing models provides a high potential in reducing modeling efforts [18].

Model Reuse

One reuse strategy is to consider standard sub-models as a black box, whose functionality may be related to input, output, control variables and decision parameters without any knowledge of its internal workings. The opposite is a white box object or system where its internal components or logic is available for inspection and modification, such as an application code. The model reuse concept may also refer to the case of modifying an existing component or full product models to fulfill purposes for which they were not originally made. Progress in model reuse requires significant

developments in several areas such as 1) understanding what information is needed to support reuse and how this should be represented, 2) developing mechanisms, automated and manual to collect and record this information, 3) understanding how to design for reuse, 4) developing analysis and search tools to locate appropriate existing components, 5) documentation of Validation and Verification (V&V) steps, which cannot be fully fulfilled [19].

Model Reuse (via modification): White Box Model

Reusing of the entire simulation model which is complex and particularly challenging for model validation. Model reuse through modification requires well-defined documentation and towering engineer knowledge about the system and the conditions of interest. The diversity of objectives for different simulation uses makes creation of models that can satisfy all intended simulation needs infeasible. On the other hand, if the user lacks critical domain knowledge, model validation can be as time-consuming and costly as developing a similar model from scratch (Model Refactoring) [4]. Possible risks and consequences of reusing models include errors-prone decision, loss of intellectual properties and difficulty to modify a model when requirements change. Thus, the knowledge encapsulated in each numerical model needs to be coherent for it to be used for different purposes. In most cases, however, model developers and users do not have the same level of understanding of the model, which may cause them to use different naming conventions, model organizations, numbers of ports, and other conventions. Not only is it a wasted time for subgroups to duplicate each other’s work, it can also introduce errors. Unless an M&S practitioner understands the model’s contextual dependencies accurately and unambiguously, model reuse (via modification) will continue to be an ineffective trial-and-error effort. Therefore, the main motivation for model plug-ins is to introduce the ‘single source of authority’ concept as a part of the model reuse strategy [20]. There exist technologies that may lower the cost of reusing a model or allow reuse of models in previously impractical situations. However, improper reuse of numerical models can undermine these gains as the consequences of a bad decision made with an invalid model can easily outweigh the benefits of reusing the model.

The open question about numerical model reuse is not only whether engineers reuse existing models properly, but how valuable they can make the practice throughout the modeling lifecycle.

Single source of authority or right from me

In the M&S reuse activity, there are likely to be at least two groups of people involved, sometimes many more (e.g., external or internal model suppliers). The basic idea is therefore to give the full modification right to the model developer (provider). It means that only the model developer is eligible to modify his/her model. Hence, the model provider is suggested to be the single source of authority in the model reuse strategy deployment [20].

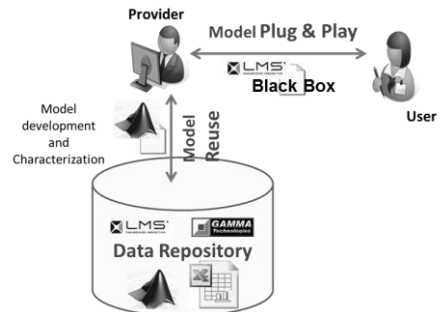


Figure 9 Model Reuse in Engineering Practice

In the literature, the term “Single Source of Authority” is used for access control to a computer system to reduce the likelihood of data being overwritten, or for other security issues. Model providers must be confident that their models are used appropriately and model user can plug-and-play the provided model. In other words, the provided model must comply with the needs of the model users [20]. The term single source of authority is consistent with a term commonly used in the lean community: “Right-from-me” [21]. Right from me, means that we should get it right the first time in all process stages from preparation of tender documents, and model to right-time delivery. In the development of fit-for-purpose instructions model, it is necessary to prevent mistakes to the greatest possible extent. When we discover any non-conformity – abnormal situation, all employees have a duty to act, correct or halt the process. Everything from mistakes in drawings to faults in the equipment – the people from the previous manufacturing stage or supplier who have caused the non-conformity must be informed immediately (real time).

Model Reuse: Black Box Model

Black box model refers here to reuse of an existing model for the same purpose for which it was originally constructed without any modifying. Model users should be able to assemble the existing model in a plug-in manner, thus minimizing the time, cost and expertise required to construct comprehensive models within the context of their organization. This is possible when a model is used on a routine basis to support tactical decision making within known and defined limits. It is not possible, however, to be sure that reuse is viable when a model is used for a purpose different from for which it is built or is used in combination with other models, possibly based on different sets of assumptions. If a model is to be reused for a purpose other than that for which it is constructed, it is vital to establish a new credibility assessment process against which the model’s validity may be assessed in its new environment of use. Assuming that the characteristics of a reused model to transfer from one provider to another, like its credibility will transfer from one application to another, are simply not justified (Fig.4) [18-20].

4 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The authors are discussed the context of reuse and a pull-type model building process (Preliminary Model Design / Virtual Prototype). The fact that we would like to use the suppliers' model as a black box fashion, it requires to check the correctness of the suppliers' models to detect any potential problem before the IVVQ (Integration, Verification, Validation, and Qualification) phase. The pull type process presented in this paper is an improvement over the traditional V-cycle because correctness check at early design phase happens and it may reduce rework in the more expensive implementation and physical prototype validation phase, which is the main driver for product development cost. This paper introduces also some important and novel terms such as single source of authority and model plug and play to highlight the traceability and intellectual properties of a simulation model. This paper does not give a concrete result in terms qualitative and quantitative improvement of current model development activity but this work proposes a robust methodology for possible improvement. The proposed methodology is supposed to be test by an industrial case study for understand its capacity. Thus the future work includes the validation protocol of proposed concepts in terms of value addition to company's current situation. Also, authors want to estimate and compare the task durations, estimated probabilities of rework and resource limitations once the overall structures of the 'as-is' and 'to-be' process models will be agreed.

REFERENCES

- [1] G. Sirin, B. Yannou, E. Coatanéa and E. Landel, "Creating a Domain Ontology to Support the Numerical Models Exchange between Suppliers and Users in a Complex System Design". Presented at the Conference on ASME (IDETC/CIE), Portland, USA, Aug., 4-7, 2013
- [2] W.L. Oberkampf *et al.* "A New Methodology for the Estimation of Total Uncertainty in Computational Simulation," AIAA 99-1612, April, 1999.
- [3] T. Vosgien *et al.*, "Lean Approach to Integrate Collaborative Product Development Processes and Digital Engineering Systems", ICED11, 15 - 18 August 2011, Denmark.
- [4] R.J. Malak, and C.J.J. Paredis, (2007). Validating Behavioral Models for Reuse. Research in Engineering Design, Vol.18, No.3, p.111-128.
- [5] N. Kohler, (2008), "Long-term design, management and finance for the built environment" Building Research & Information; March, Vol. 36(2), pp.189-194.
- [6] Systems Engineering Fundamentals, DOD Supplementary text, Belvoir, Virginia 22060-5565, USA, 2001. Available:http://123management.nl/0/070_methode/072_kwaliteit/Dod%20Systems%20Engineering.pdf
- [7] J. Hagel and J. S. Brown, "From Push to Pull-Emerging Models for Mobilizing Resources", Working Paper, October 2005, available:http://www.johnhagel.com/paper_pushpull.pdf
- [8] "The Guide to Lean Enablers for Managing Engineering Programs", Version 1.0, Published by the Joint MIT PMI INCOSE Community of Practice, May 2012.
- [9] E. Rebentisch, D.H. Rhodes, E. Murman, "Lean Systems Engineering: Research Initiatives in Support of a New Paradigm", Conference on Systems Engineering Research (CSER), Los Angeles, USA, 2004.
- [10] O. De Weck, "Feasibility of a 5x Speedup in System Development due to Meta Design". DETC2012-70791, 2012.
- [11] L. McGinnis, and V. Ustun, 2009, "A Simple Example of SysML-Driven Simulation," Proceedings of the 2009 Winter Simulation Conference (WSC), IEEE, pp. 1703-1710.
- [12] A. Van der Velden, P. Koch, S. Devanathan, J. Haan, D. Naehring and D. Fox, "Probabilistic Certificate of Correctness for Cyber Physical Systems", Volume 2: 32nd Computers and Information in Engineering Conference, Parts A and B Chicago, Illinois, USA, August 12-15, 2012.
- [13] S. Friedenthal, A. Moore and R. Steiner, "A practical guide to sysml: The systems modeling language", 2011.
- [14] J.P. Womack and D.T Jones, "Lean Thinking. Banish Waste and Create Wealth in Your Corporation", 2. Edition. New York: Simon and Schuster, 2003.
- [15] C. Bauch, "Lean Product Development: Making waste transparent", Diploma thesis from TUM and MIT, August 2004.
- [16] J. Kato: Development of a Process for Continuous Creation of Lean Value in Product Development Organizations. Department of Mechanical Engineering, MIT, 2005.
- [17] J. M. Branscomb, C. J. Paredis, J. Che and Jennings, M. J., 2013, "Supporting Multidisciplinary Vehicle Analysis Using a Vehicle Reference Architecture Model in SysML," Procedia Computer Science, 16, pp. 79-88. available:http://dspace.mit.edu/bitstream/handle/1721.1/81429/TH_Bauch_04.pdf?sequence=1
- [18] L. Yilmaz. (2004). On the Need for Contextualized Intropective Models to Improve Reuse and Composability of Defense Simulations, JDMS Volume 1, Number 3.
- [19] A.S. Mohamed, S. Khaled and E. Aziz, "Component Based Modeling and Simulation of Value Stream Mapping for Lean Production Systems", Department of Industrial and Management Engineering College of Engineering and Technology Arab Academy for Science, Alexandria, Egypt.
- [20] J.D. Moffett and M.S. Sloman "The Source of Authority for Commercial Access Control", IEEE, 0018-9162/88/0200-0059\$01.00 February, 1988.
- [21] Konstructions - Bakelit AB, KB Lean Enterprise, "Right from me". Available:http://kbcomponents.com/wp/wpcontent/uploads/2012/09/KB-0001_Leanhandbok_ENG_LoRes.pdf
- [22] J. Oehmen and E. Rebentisch, "Waste in Lean Product Development", Version 1.1, July 2010, available: <http://dspace.mit.edu/handle/1721.1/79838>