On aircraft simulation in conceptual design

Alexandra Oprea
This is a Swedish Licentiate’s Thesis

Swedish postgraduate education leads to a doctor’s degree and/or a licentiate’s degree. A doctor’s degree comprises 240 ECTS credits (4 years of full-time studies). A licentiate’s degree comprises 120 ECTS credits.

Typeset using \LaTeX

Printed by LiU-Tryck, Linköping 2022

Edition 1:1

© Alexandra Oprea, 2022
https://doi.org/10.3384/9789179294762
ISSN 0280-7971
Published articles have been reprinted with permission from the respective copyright holder, with minor adaptations

Front cover: artwork based on JAS39 photos; original photos copyright Saab AB
To Astrid and Matic.
Abstract

The aerospace industry has a long tradition of using Model Based Systems Engineering in its development processes. Throughout the years an extensive library of system simulation models has been created, expanding with every new model being developed. Nevertheless, modelling and simulation engineers prefer developing their own models from scratch, as the reuse of legacy models seems cumbersome, leading to the simultaneous existence of multiple simulation models of the same system in a project. Not only does the development, verification and validation and maintenance of these multiple simulation models incur extra costs, but they also represent a potential threat to the aircraft development, as data consistency is more difficult to be ensured.

The aim of this thesis is to investigate simulation model reuse during the aircraft development process, with the goal of creating an easy-to-reconfigure simulation framework, allowing for the inclusion of simulation models of various levels of fidelity. Through an industry-as-laboratory approach, the inclusion of industry-grade legacy system simulation models is investigated, with the help of simulation model integration standards such as Functional Mock-up Interface and System Structure and Parametrisation.

The results of the work include the development of a proof-of-concept simulation framework that enables the aggregation, integration and reuse of simulation models from different technological domains and with different levels of fidelity. Moreover, the work proposes a workflow that streamlines the decision milestones when faced with the possibility of selecting between creating a new system simulation model or reusing a legacy one, and highlights the challenges associated with model reuse from both a technical and an organisational perspective.
Acknowledgements

First of all, I would like to extend my deepest gratitude to my supervisor, Dr. Hampus Gavel, for placing a great amount of confidence in me from Day One. Thank you for all the discussions, the guidance, and for keeping this project anchored to reality whenever it drifted off on unfeasible paths.

I would also like to thank my team of co-supervisors: Dr. Ingo Staack, Dr. Magnus Eek and Prof. Johan Ölvander, for their efforts. Your expertise and your different perspectives helped me escape my industry-set ways and ultimately led to better research.

A special thank you goes out to my Saab managers (in order of appearance): Daniela Svensson, Anton Jonzén, Marina Wallin and Dr. Klara Grönhagen: I am grateful that you recognised the potential in this work and protected it from being buried under other industrial assignments.

I would also like to thank Saab and VINNOVA for the financing of this research through the 7th National Aeronautics Research Programme.

As an Industrial PhD student I am fortunate enough to have two sets of colleagues to be thankful for: the colleagues from Saab, especially the Division of Flight Mechanics and Aircraft Performance, with a special mention to Dr. Andreas Nilsson, who has helped me make sense of the future for aircraft simulation; and my colleagues at Linköping University, from both the Product Realisation and the Flumes departments. I should not forget all the PhD students I’ve met along the way. This journey was long and winding, but in your company I really enjoyed the ride!

And, in the end, I would like to express my gratitude to my family: my parents, Ana and Eugen, my sister Roxana, and, last, but definitely not least, my partner Matic, and my daughter Astrid. Your names should be next to mine under the title, because this work would have not been possible without your support.

Alexandra Oprea
Linköping, 2022
Appendixed Papers

The following papers have been reprinted as they have appeared in the original publication, with minor modifications in formatting. They will be referred to as Paper I - III in the text.

For Papers I and III the first author is the main author, responsible with experiment planning and execution, results interpretation and manuscript writing, with input from the rest of the authors. For Paper II the first two authors are the main authors, with input from the rest of the authors.


Acronyms

ACD  Aircraft Conceptual Design.
API  Application Programming Interface.
ARES Aircraft Rigid body Engineering Simulation.
CAD  Computer-Aided Design.
CAS  Credibility Assessment Scale.
CFD  Computational Fluid Dynamics.
FMI  Functional Mock-up Interface.
FMU  Functional Mock-up Unit.
HIL  Hardware-In-the-Loop.
M&S  Modelling & Simulation.
OD  Operational Domain.
ODE  Ordinary Differential Equation.
PDP  Product Development Process.
PIL  Pilot-In-the-Loop.
SIL  Software-In-the-Loop.
SISO  Simulations Interoperability Standards Organization, Inc..
SoTA  State-of-the-Art.
SSD  System Structure Description.
SSP  System Structure and Parametrisation.
SSV  System Structure Values.

V&V  Verification and Validation.

XML  EXtensible Markup Language.
# Contents

1 Introduction .................................. 3  
  1.1 Aim ........................................... 5  
  1.2 Method ........................................ 6 
  1.3 Definitions and delimitations ..................... 7 
  1.4 Thesis structure ................................ 8 
  1.5 Contributions .................................. 9 
  1.6 Research funding ............................ 10 

2 Development of simulation models ............................. 11  
  2.1 The product development process ................... 12  
  2.2 Models and simulators ............................ 13  
  2.3 Model development processes ....................... 15  
  2.4 Model reuse ................................... 18  
  2.5 Simulation model properties ....................... 19 

3 Standards for model and simulation integration .......... 25  
  3.1 The Functional Mock-up Interface standard .......... 26  
  3.2 The System Structure and Parametrisation standard .. 27 
  3.3 Benefits of using open standards ................... 27 

4 Reuse of system simulation models in Aircraft Conceptual Design 29  
  4.1 The conventional Aircraft Conceptual Design process .... 31 
  4.2 The proposed Aircraft Conceptual Design framework ..... 32  
  4.3 Legacy model integration .......................... 33 
  4.4 Flowchart for simulation model creation ............... 36 

5 Discussion and Conclusion ............................... 39  
  5.1 Overview of the appended papers .................... 39  
  5.2 Discussion ...................................... 40  
  5.3 Conclusion ...................................... 44 
  5.4 Future work ..................................... 45 

Bibliography ........................................... 47

xiii
Introduction

Modelling & Simulation (M&S) is a concept that has penetrated all aspects of the aerospace industry. From solid, geometrical models in Computer-Aided Design (CAD), and Computational Fluid Dynamics (CFD) analysis for aerodynamics, to mathematical models for essentially everything aircraft-related, it has become a methodology that enables increased product complexity. However, model development comes at a cost, and modern jet fighter programmes have been plagued by cost and time overruns. Any method that could get a more complex product on the market without unnecessary delays is of considerable benefit to a manufacturing company.

The development of any new product, aircraft included, is generally performed according to a Product Development Process (PDP). The PDP consists of well-defined stages covering all the development activities from early concept to system development, testing and production. Several disciplines are involved in the process. For instance, the marketing department performs the market research identifying the needs of the product, the technical departments perform the technical development, the financial department is involved in the cost estimations and so on. Close communication is required at all stages and between all domains to achieve a harmonious development.

Saab Aeronautics develops products according to ISO/IEC 15288 ¹ [1, 2]. M&S is used at all stages of the PDP and in all engineering domains. From a technical

¹ISO: International Organisation for Standardisation; IEC: International Electrotechnical Commission
1. INTRODUCTION

development perspective, at the early conceptual design stage, for a given set of requirements, multiple engineering teams draft concepts for their own subsystems, more or less independently of each other. When a final geometry for the aircraft concept is chosen, the concept advances into the development stage, where the design work continues for each of the different engineering teams. Although everyone is working towards the same goal and has access to the same data, the systems development work is performed intra-disciplinarily, and cross-disciplinary impact is assessed mainly through the exchange of data and assessments of results. This can lead to design decisions that turn out to be detrimental to the project down the line. This phenomenon, called the Design Paradox, captured by Verhagen [3] and paraphrased by others [4], is shown in Figure 1.1. In general terms, during a project, decisions need to be taken in the beginning, when there is little information about the product (be it a system, subsystem, system of systems, etc.). These decisions also incur committed costs, as development processes tailored to the chosen design are initiated and testing rigs are being set up. Therefore, although the product is still in its initial stage, a considerable part of its cost has already been allocated, without having a corresponding amount of information available. With time, as design progresses, more information becomes available about the product; however, the design freedom is limited by the choices made in the early stages of the product’s development. Therefore, the more information there is available at the early stages of the PDP, the better the decisions will be regarding the product and its subsystems. The goal is to minimise the risk of redesign at later stages, when doing so is too costly.

For an aircraft-specific application, increasing the level of information can be done through several methods, like early prototype building of several configurations, wind tunnel testing or sub-scale flight testing [5] or others. However, a simple, relatively cheap method is the reuse of existing simulation models from other projects. Doing so would shift focus from simulation model development as one of the primary activities in early concept development (with all the steps that it includes, namely data acquisition, model building, Verification and Validation (V&V), etc.), to model use. This paradigm shift sets new requirements on the simulation model development process, M&S governance and the aircraft development process as a whole. The reuse of high-fidelity models allows for technological exploration earlier in the development process, and therefore enables earlier discovery of design interactions and potential problems. This poses requirements on the collaboration and information exchange processes within the project and the company, not only from a technical point of view, but also from a project management perspective, where everyone is aware of the current goals, which can be fluid in the beginning of any project.

Different project stages require different simulation models providing different types of information. The traditional V-model [1], shown in Figure 1.2, can be used to provide an overview of the types of simulation models that are required and when
1.1. Aim

The purpose of this thesis is to investigate simulation model reuse during the aircraft development process, with the goal of creating a fast, easy-to-reconfigure
1. INTRODUCTION

Figure 1.2: The Development V, adapted from [1]

simulator, allowing for varying levels of simulation model fidelity. The reconfiguration term refers to the possibility of customising the simulator for different aircraft or aircraft systems without the need to change its fundamental structure. This aim is broken down into two research questions:

RQ1 How can an aircraft simulator be designed to enable aggregation, integration and reuse of simulation models from different technological domains, with different levels of fidelity?

RQ2 What are the challenges encountered when reusing legacy system simulation models in new development projects?

Aircraft simulation is a multidisciplinary field, and RQ1 aims to tackle the challenges that are common to all the disciplines involved. One of the main problems identified, the reusability of simulation models, opens up for RQ2, where a suitable simulator architecture that allows for simulation model reuse is examined. However, having the technical possibility to interconnect simulation models does not guarantee trustworthy simulation results, and therefore RQ2 also investigates the challenges encountered when reusing legacy system models, whose intended use does not match entirely the intended use of the new system under development.

1.2 Method

The work follows the method described in the industry-as-laboratory ([6], [7]) methodology, shown here in Figure 1.3. This method is well-suited for projects that require collaboration between the industry and academia, as it combines real-life problems, stemming from industry, with rigorous academic research. The devised solutions are implemented and evaluated in the industry.

The overarching theme of increased product knowledge through simulation model reuse from the research questions stems from the industry, here seen as a
1.3 Definitions and delimitations

source of inspiration. Through research, the State-of-the-Art (SotA) in academia is assessed (discussed in Paper I), and a hypothesis for possible solutions is formulated. A "proof-of-concept" solution is then implemented and evaluated, and its results are assessed from an industrial perspective, using the industry as an application playground, through the provision of industry-grade legacy system simulation models (through Papers II and III).

1.3 Definitions and delimitations

This thesis is concerned with M&S for aeronautics and vehicle systems development. Therefore, other types of M&S have not been taken into consideration (structural analysis, radar signature calculations, human-machine interaction, communications, etc.). This does not imply that the results are not applicable to these domains, but that the author has not been concerned with the precise evaluation of the extent of this applicability.

Moreover, although the thesis is set in an aeronautics context, other industries could also benefit from the results, namely any industry that uses M&S in their development processes and is confronted with the challenges of legacy model reuse. The industrial perspective might also be useful to academia, especially in the fields developing simulation software that outlasts the time limitations incurred by the nature of doctoral students and researchers employment type. However, as academic research is driven by different goals than industrial research, there might be other aspects that come at play. These have not been investigated at all in this work.

The terms "model", "simulation" and "simulator" and their derivatives will be used frequently throughout the thesis. To avoid confusion, their definition in the context of this work is required.
1. **INTRODUCTION**

**Definition 1.** A "model" of a system is a mathematical representation of the system properties of interest.

According to Ref. [8], "a model of a system is a tool we use to answer questions about the system without having to do an experiment". For mathematical models, "the relationships between quantities [...] that can be observed in the system are described as mathematical relations in the model".

**Definition 2.** Simulation: Experimentation with the model, through the application of input data in order to receive output data, increasing knowledge about the model or its target system.

**Definition 3.** Simulator: A model or collection of models and the required infrastructure that allows the execution of simulations.

The term "simulator" requires special attention. In the context of aircraft simulation, this term usually leads to the mental image of an aircraft cockpit mock-up allowing flight enthusiasts to experience flying without leaving the ground. However, the simulators this thesis is concerned with do not require the presence of a pilot, i.e., a human-in-the-loop. The results, however, are applicable to any type of simulator.

1.4 **Thesis structure**

This thesis is structured into five main sections:

1. Introduction
2. Development of simulation models
3. Standards for model and simulation integration
4. Reuse of system simulation models in Aircraft Conceptual Design
5. Discussion and Conclusion

The current section, Introduction, sets the background for the current project. Section 2, Development of simulation models, presents the theory behind aerospace simulation models, to the extent relevant for this work. Section 3, Standards for model and simulation integration, presents the Functional Mock-up Interface (FMI) and System Structure and Parametrisation (SSP) standards that are used in this thesis. Section 4, Reuse of system simulation models in Aircraft Conceptual Design, presents the work performed in this project and is based on the contents of the appended papers I, II and III. Section 4, Discussion and conclusion, provides an overview of the results and aims to place the work in a larger context.
1.5 Contributions

As the thesis constitutes a compilation of papers, the main contributions are delivered through the appended papers, whereas the first part of the thesis defines the background to the research and highlights how the papers are interconnected. The papers are appended in the order of publication. Paper I addresses the academic SotA of the problem coming from the industrial environment. It identifies appropriate methodologies for simulation model development and standards for simulation model integration, essential for the successful development of the aircraft simulator that is sought in RQ1. In Papers II and III the integration standards are explicitly used for the development of a proof-of-concept framework that fulfils the criteria from RQ1, whereas the results referring to model development from Paper I are implicitly used when assessing the usability of the framework in an industrial context. RQ2 is handled in Paper III, where the benefits and challenges of introducing an industry-grade legacy system simulation model in the existing framework from Paper II are assessed. Table 1.1 provides an overview of how the two RQs are handled in the papers. A more thorough description of the appended papers is given in Section 5.1.

Table 1.1: Overview of the handling of the RQs by the appended papers

<table>
<thead>
<tr>
<th></th>
<th>Paper I</th>
<th>Paper II</th>
<th>Paper III</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>[x]</td>
<td>[x]</td>
<td>[x]</td>
</tr>
<tr>
<td>RQ2</td>
<td></td>
<td>[x]</td>
<td></td>
</tr>
</tbody>
</table>

1.5 Contributions

At an abstract level, the work brings an industrial perspective on the implementation of tools, standards and sets of best practices advocated in academia or single disciplinary product development. A holistic view on aircraft simulation has been taken, as the work positions itself at the intersection of multiple technical disciplines (Aircraft Conceptual Design (ACD), aeronautics, vehicle systems and even software development) and crosses into the domain of management by bringing up questions regarding collaboration workflows or company culture and their influence on the multidisciplinary development of an aircraft.

On a technical level, a proof-of-concept of a simulation and optimisation framework is developed using the latest integration and simulation standards. The framework is then used to integrate industry-grade legacy system simulation models for the development of new aircraft concepts. As the technological and implementation difficulties are overcome, their implications come under scrutiny: just because a higher fidelity model can be integrated earlier in the design process, it does not mean it should be, and, likewise, just because one can build another simulation model, it is not always the right approach. The work raises the need for further investigation on
1. INTRODUCTION

the topic of simulation model fidelity, validity and information compatibility, as the model information content has a direct impact on the outcome of concept generation.

1.6 Research funding

The research presented here has been funded by VINNOVA\(^2\) and Saab Aeronautics and is included in the NFFP7\(^3\) Flexible Simulation Framework for Flight Mechanics project (grant 2017-04889). Papers II and III have been developed in collaboration with the NFFP7 Digital Twin for Automated Model Validation and Flight Test Evaluation project and the ITEA3\(^4\) project EMBrACE.

---

\(^2\)VINNOVA - Sweden’s Innovation Agency. www.vinnova.se  
\(^3\)NFFP7 - National Aeronautical Research Program 7  
\(^4\)ITEA - Information Technology for European Advancement (The Eureka RD&I Cluster on Software Innovation); www.itea3.org
2 Development of simulation models

A modern aircraft is a conglomerate of highly integrated systems with different natures, all subjected to strict requirements concerning weight, volume, consumed power, reliability, maintainability, cost, etc. To give an example, Figure 2.1 shows the Saab-developed Gripen E, test aircraft 39-9, during taxi. Several systems can be observed: the extended landing gear, facilitating the movement on the ground, the fuel system through the external fuel tank in a central hardpoint, the radar in the nosecone, the air inlet for the engine, the external stores on the wing hardpoints, some of the control surfaces (canards, the rudder), the pilot and so on. All these systems are interconnected. For instance, they all require electrical power to function, even the pilot, who is dependent on the Environmental Control System for cockpit pressurisation and on the anti-g suit to survive. Should one of the systems require more power, then other consumers need to be redesigned for a reduced power consumption, as the total available is limited. Similarly, should the aircraft require increased control surfaces for manoeuvrability purposes, then less wing volume is available for fuel storage, impacting the range of the aircraft. In the early days, systems were more federated than today’s systems and development relied more on experimental testing. Nowadays, systems have a much higher degree of integration in order to save weight; this is enabled by the use of M&S to understand the increased complexity. This chapter intends to give an overview of how M&S is related to the aircraft development process, presented as a general PDP, and introduce the concepts related to M&S that are of interest in this thesis.
2. DEVELOPMENT OF SIMULATION MODELS

Figure 2.1: JAS 39 Gripen E Test aircraft 39-9 during taxi. Photo: Linus Svensson, Saab AB

2.1 The product development process

As mentioned in Chapter 1, designing a new aircraft at Saab Aeronautics follows the system life cycle processes from ISO/IEC 15288:2015 Systems and software engineering - System life cycle processes, detailed in [1] and [2]. This thesis is related to the technical side of the PDP, which is covered by multiple authors in the literature [9, 10]. Ulrich and Eppinger’s version of the process [10] is given as an example. The process consists of six stages:

1. Planning
2. Concept Development
3. System-level Design
4. Detail Design
5. Testing and Refinement
6. Production Ramp-up.

The first stage of the process translates the requirements coming from the stakeholders into a Design Space. Several concepts that fit the requirements are evaluated, with one being chosen at the end of the Concept Development Stage. Next, the development of the concept systems is initiated. The details of the design are set in the Detail Design stage. Then, the product undergoes testing, after which production starts, and ultimately the product is deployed into market. Even if this idealised process is described in a linear manner, in practice it is highly iterative,
2.2 Models and simulators

as every stage requires several iteration steps until all the requirements are met and the development can move on to the next stage. However, moving from the late development stages back to the drawing board of the first stages incurs high costs that can even jeopardise the success of the project.

As this thesis is concerned with the concept development stage, it will be described in more detail. Figure 2.2, from [10], depicts the concept selection process. In an iterative manner, concepts are generated, developed, improved and/or discarded, until only one is left which will proceed to the next stages of the development process.

![Figure 2.2: Eppinger’s concept generation and selection process, from [10]](image)

The Design Paradox from Figure 1.1 presents a theoretical view of, among other things, the costs incurred at the different stages of a project. In Incose’s Systems Engineering Handbook [1] these theoretical costs are quantified, based on a statistical analysis of projects from the US Department of Defense reported by the Defense Acquisition University [11] and presented here in Figure 2.3. A quantification of the allocated costs for each design stage is performed, represented by the grey boxes in the figure. For the concept stage, 8% of the costs are spent, while during design an extra 7% of the costs is spent, totalling 15% of allocated costs. Another 5% is spent on development, totalling 20% of allocated costs, with a further 30% spent during production and testing, totalling 50%. The remaining 50% of the total life-cycle cost is spent in operations through disposal. At the same time, 70% of the committed cost is achieved already by the end of the concept phase. It should be noted that the quantification of the costs in Figure 2.3 was performed in 1993, and the values for each staple (Concept, Design, Develop, Production and Testing, and Operations through Disposal) might not be the same today; however, the principle of having committed most of the cost in the early stages of the project still stands.

2.2 Models and simulators

Through M&S more information about the aircraft can be gathered during the first stages of the development process, so that late redesigns can be avoided and the
2. DEVELOPMENT OF SIMULATION MODELS

Figure 2.3: Committed life cycle cost against time. From Incose [1], based on [11]

allocated 70% of the costs at the end of the concept stage represent less of a risk. The simulation models evolve as the aircraft design matures, as every development stage adds different requirements on the existing simulation models. Most of the time, the same system is represented by several simulation models depending on which information the simulation model is supposed to provide. For instance, an engine system simulation model can be represented by a multi-domain equation-based model in the propulsion development group and by simple tabulated maximum thrust values depending on altitude in others. Several simulation models can be assembled into simulators, ranging from simple desktop simulators of, for instance, a system and its control software, to complete aircraft system simulators that include hardware, software and a human in the loop. In general, simulators can be divided in three categories:

1. small-scale simulators
2. mid-scale simulators
3. large-scale simulators.

Hällqvist [12] presents a thorough description of these three simulator types, as they are implemented at Saab Aeronautics. A shorter version of it is presented here. Small-scale simulators are used for the development of a standalone subsystem. They are made of software models, and they include models from at least two engineering disciplines. An example of a small-scale simulator would be an aircraft subsystem hardware simulation model connected to its controlling software. Mid-scale simulators are used for the development of a few aircraft subsystems. They typically include models developed in different tools that are simulated together in
A third-party tool. Finally, large-scale simulators include most aircraft subsystems, either as software-based models, Hardware-In-the-Loop (HIL), or a combination of both [12–14].

Aircraft simulators can also be classified by their component types:

• Software-In-the-Loop (SIL);
• Hardware-In-the-Loop (HIL); and
• Pilot-In-the-Loop (PIL);

or by their purpose:

• training simulators for pilots;
• training simulators for maintenance personnel;
• tactical simulators; and
• development simulators.

An SIL simulator includes only software-based models. An HIL simulator, on the other hand, includes hardware components, that can function both alongside software components and without them. A PIL simulator requires the presence of a pilot to close the control loop. Training simulators are, as the name suggests, for training purposes, both for pilots and maintenance personnel. Tactical simulators can be used when evaluating the tactical strategies of entire air units. The development simulators are used, as the name suggests, during the entire aircraft development process. The above lists are not exhaustive, however; the intent is to demonstrate that simulators can have very different purposes, which impacts the simulation models or hardware components from which they are made.

Irrespective of a simulator’s category, its constituting unit is a system simulation model. A simulation model can be either hardware or software. This work focuses only on simulation models implemented as software, here seen as mathematical representations of the underlying physics of a system, whether through physical equations or mathematical approximations of these.

2.3 Model development processes

In Paper I the assessment of the SotA for the development of flight simulation models was performed. It was discovered that developing a simulation model usually follows the process in Figure 2.4. There is a problem to be solved, and a sketch of a solution is implemented. The solution is continuously tested and improved until a satisfactory level of the solution assessment criteria is reached. This process can
be compared to the development processes encountered in commercial software development [15], presented in Figure 2.5. In the commercial software development methodologies, one starts with the analysis of requirements and stakeholder needs, and the solution is developed according to a process best fit for the application, either sequential, incremental or, most often, a combination of both [15]. The developed software is a product in itself, a commodity, and its development is tightly coupled to budget and time constraints associated with the software project. In contrast, the models created for systems development have as a sole purpose to create knowledge about the system of interest. Although time and budget constraints impact their development, they are not the main development drivers; the model is ready when the systems engineer is satisfied with its output. This type of software is called scientific software and is defined by Kelly [16] as "software that includes a large component of knowledge from the scientific application domain and is used to increase the knowledge of science for the purpose of solving real-world problems." According to Kelly, scientific software is characterised by the following features:

1. its development requires the involvement of a domain specialist;
2. the software user has knowledge of the application domain, allowing for correct result interpretation;
3. the sole recipient of the software output is the user; the software is not used to control equipment;
4. the sole purpose of the software is to advance knowledge about a phenomenon / a real-world problem;
5. the most important software quality is correctness, "to the exclusion of all else". [16]

![Diagram of model development process](image)

Figure 2.4: The process of model development, according to Johanson and Hasselbring [17]

If one were to single out the most important characteristic of a model during aircraft design it would be the fourth characteristic (in bold) from the above list:
2.3. Model development processes

Figure 2.5: The phases of the commercial software development life cycles, adapted from Ref. [15]

the purpose of a model is to create knowledge about the system it represents. The main challenge in doing so, when modelling for aerospace applications (or any other scientific-intense applications, like nuclear engineering, weather prediction etc.) is that the complete list of requirements for the solution is rarely known beforehand, see Ref. [16–22]. In practice, this initial lack of fixed requirements leads to a modelling solution being drafted to fit the list of requirements known at that specific point in time. Once the simulation model is satisfactory, it is used for generating information. With time, the requirements list changes, as either the scope of the project is extended or modified. The already-built simulation model is then adjusted to fit the new scope. With every new requirement change new adjustments need to be made, which in the end might lead to sub-optimal solutions that are resource consuming to maintain and reuse.

One way of mitigating the risk posed by incomplete or unknown requirements is to implement a mix of the two processes mentioned above. This process is illustrated in Figure 2.6 and implemented at Saab Aeronautics. It is described in detail in Ref. [23, 24]. This process can be regarded as an adaptation of the rigorous, formal practices of the Software Engineering domain to the reality of the Swedish aeronautics industry.

The workflow starts with the definition of the intended use of a model. The arrow-shaped blocks represent the activities that the model undergoes; the green blocks are the outcome of the activities (1 - the simulation model itself, 2 - the simulation model interface description, 3 - relevant test cases, 4 - generated model code to be used in higher level simulators and 5 - simulator integration layers). The blue circles at the bottom side of the figure are the required documentation for the model, including a Requirement Specification "RS" and a Status Declaration "SD". At the end of the process, the existing models are well documented. They can, and
2. DEVELOPMENT OF SIMULATION MODELS

![Diagram of Development of Simulation Models Workflow at Saab Aeronautics, from [23]](image)

should, provide a starting point whenever the need for a similar model arises. This workflow is iterative, and it should be adapted to the particularities of every project. For smaller projects, of a more informal nature and without strict requirements on a structured workflow, the steps from Figure 2.6 turn into the ones from Figure 2.4.

2.4 Model reuse

Model reuse could be a key contributor in the chase for better design solutions, with fewer to no redesigns (and, ultimately, better aircraft). There are three types of model reuse:

1. between products;
2. between projects; or
3. within the same project.

To define the three categories, an example of a landing gear system simulation model is taken. In the first case, reuse between products, the same landing gear system simulation model could (in an idealised case) be used for the development of a military transport aircraft or an airliner, provided the systems and the simulation model properties (see Section 2.5) are similar. These are two different products, with different requirements, yet simulation model reuse can be achieved. The second category, reuse between projects, addresses the case where families of products are developed. In this case the requirements are similar, so reuse is possible (and even advisable). The third category, reuse within the same project, addresses the need to simulate the same system for different disciplines in the context of the same project. For instance, the landing gear simulation model is required both at the department concerned with developing the landing gear and at the hydraulics department.
Model reuse between two very different products, for instance a system used on both a fighter aircraft and a demonstrator aircraft for research projects, may involve administrative or juridical hinders. This is not a situation handled in this thesis, albeit from a technical point of view it should not be too different from model reuse between (similar) projects.

Model reuse within the same project implies either the exchange of simulation models between different engineering domains or the reuse of legacy simulation models within the same engineering domain, however altered to fit the current project specifications. Ideally, when the need occurs for a new system simulation model, the task is delegated to the responsible model or system owner. They then decide whether model reuse is possible or a new model needs to be developed. In practice, however, depending on the workflows at the company, the management structure, the complexity of the model, the time and resources available and the modelling engineers’ knowledge, the model might be created at the department that needs it, without verifying whether the model already exists. This situation leads to:

- an unnecessarily high number of models for the same systems scattered around the company;
- data inconsistency;
- lack of traceability; and
- extra time consumed on creating, maintaining, and integrating these models.

One of the reasons the above occurs is that domain-specific simulation models are usually created in tools that are incompatible with tools from other domains, and therefore a re-creation of the simulation model in a different tool is deemed a "faster" solution than the integration of the existing model into an incompatible tool. Ideally, if a model exists that fits the intended use and the Operational Domain (OD) (defined in the section below) and is of an adequate level of fidelity, simply "take it" and plug in one’s own simulation framework. To achieve this goal, model integration standards have been developed, see Ref. [25–29]. The work in this thesis is based on the Functional Mock-up Interface (FMI) and System Structure and Parametrisation (SSP) standards [26, 27], thus a short description of these will be given in Chapter 3.

### 2.5 Simulation model properties

The properties of a simulation model are many and diffuse to quantify. A subset of these, relevant to this thesis, consists of a simulation model’s level of *fidelity*, its *credibility*, its *intended use*, and its *operational domain (OD)*. The model undergoes verification and validation (V&V) activities. The properties presented listed here do not form an exhaustive properties list, nor does the work in this thesis directly concern all of them. They are, however, essential for the ideas put forward in this
2. DEVELOPMENT OF SIMULATION MODELS

thesis, so a brief description of their meaning and implications is required.

Model fidelity is a term that has not reached a consensus in academia, as its meaning is tightly related to the particularities of the discipline of the model. The Simulations Interoperability Standards Organization, Inc. (SISO) defines it in Ref. [30] as "the degree to which a model or simulation reproduces the state and behaviour of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation: faithfulness." Most of the time, fidelity is mentioned as an inherent property of a particular model or simulation, usually accompanied by the vague terms "low", "medium" and "high", without any further definition or quantification. Despite its ambiguity, fidelity is identified by Roza in Ref. [31] as being "one of the main cost drivers of any model or simulation development" with higher fidelity models requiring more time and resources to achieve and/or simulate. It is, therefore, crucial to any development project to understand the required level of fidelity for the models used at all stages in the development process.

The intended use of a model is a self-explanatory term. In Ref. [32] Balci provides a detailed description of what the intended use of a model should contain as well as the impact a well-described intended use has on its use and potential accreditation. In short, the intended use is the purpose of the model together with the conditions under which the simulation model should be used.

The Operational Domain (OD) of a model is defined by Hällqvist in Ref. [12] as the n-dimensional volume spanned by all feasible values of model inputs affecting observable system states. It is important to note that the simulation model OD does not need to be (and usually does not) span the entire OD of the system it represents.

According to Balci [33], model verification is concerned with assessing whether the model was built right and model validation is concerned with assessing whether the right model is built. A definition of credibility is adapted from Pace [34]: it is the confidence that can properly be placed in the experience or results of a simulation application. A thorough overview of how verification, validation and credibility are connected is given by Hällqvist in Ref. [12] and Eek in Ref. [23]. It can be said that, strictly from a modelling perspective, the goal of any modelling activity is to create a credible simulation model within its domain of validity. To achieve that, a model must undergo V&V activities and, according to Pace, assessing model fidelity is one of the necessary steps for validating a model.

Figure 2.7 presents an overview of how the concepts mentioned above are related in the context of this thesis. The credibility of a simulation model is defined by its V&V status, its use history and, depending on the application, other properties. The V&V status is defined by the model's fidelity and its intended use. Fidelity is defined by the level of realism of the simulation model, which can either be defined...
2.5. Simulation model properties

Figure 2.7: Relations between the concepts related to simulation model properties in the context of this thesis

formally, through the historical usage of the model or formal model reviews, or informally, through face validation (between two or more engineers who agree that the model represents what it is expected to represent). This is the most frequent form of informal methods used to define the level of realism, but the author does not exclude the possibility that there might exist others. A simulation model’s level of detail can also vary. A simulation model can be a test dummy (for instance a 3D box with the same volume as the real system in a CAD representation, or a software model with the correct interface to the surrounding framework but with incorrect behaviour; a 1D representation of an object or of a phenomenon, such as fluid flow; or a static or dynamic representation of a phenomenon. Nevertheless, there is no ranking of the categories; a formal reviewed model is not necessarily more realistic than one with a historically mandated usage.

Other relations between the concepts are possible, and they might be just as valid, but this is the approach taken here. Model fidelity consists of how well a model represents the target (level of realism/representativeness) and how detailed it is in doing so, from the modelling assumptions to the way the model is built. Often, a higher level of detail leads to a higher level of realism, but this does not have to be the case. The author feels that these two terms are usually used interchangeably, but it is the author’s opinion that fidelity consists of both.

The concept of the credibility of a model is best exemplified by NASA’s Credibility Assessment Scale (CAS). This scale has been developed to “include a standard
2. DEVELOPMENT OF SIMULATION MODELS

method to assess the credibility of the M&S presented to the decision maker when making critical decisions” [35]. The scale takes eight factors into consideration. These factors cover the M&S development, the M&S operations and other supporting evidence for model credibility and can take scores from 0 to 4. They are listed below (from [35]), with the assessment criteria summarised in Table 2.1. A concept for credibility assessment based on the CAS (among others) has been developed, implemented and tested at Saab Aeronautics [36].

- **M&S Development**
  - Verification: Were the models implemented correctly, and what was the numerical error/uncertainty?
  - Validation: How well did the M&S results and the referent data compare?

- **M&S Operations**
  - Input Pedigree: How confident are we of the current input data?
  - Results Uncertainty: What is the uncertainty in the current M&S results?
  - Results Robustness: How thoroughly are the sensitivities of the current M&S results known?

- **Supporting Evidence**
  - Use History: Have the current M&S been used successfully before?
  - M&S Management: How well managed were the M&S processes?
  - People Qualifications: How qualified were the personnel?

At an overall level, it is the intended use of a simulation model that determines how thorough the credibility assessment needs to be. Simulation models that lie as a certification basis require an extensive credibility assessment, whereas some simple simulation models might not require any at all.
<table>
<thead>
<tr>
<th>Level</th>
<th>Verification</th>
<th>Validation</th>
<th>Input Pedigree</th>
<th>Results Uncertainty</th>
<th>Results Robustness</th>
<th>Use History</th>
<th>M&amp;S Management</th>
<th>People Qualifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Numerical errors small for all important features</td>
<td>Results agree with real-world data</td>
<td>Input data agree with real-world data</td>
<td>Non-deterministic &amp; numerical analysis</td>
<td>Sensitivity known for most parameters; key sensitivities identified.</td>
<td>De-facto standard.</td>
<td>Continual process improvement.</td>
<td>Extensive experience in and use of recommended practices for this particular M&amp;S.</td>
</tr>
<tr>
<td>3</td>
<td>Formal numerical error estimation.</td>
<td>Results agree with experimental data for problems of interest.</td>
<td>Input data agree with experimental data for problems of interest.</td>
<td>Non-deterministic analysis.</td>
<td>Sensitivity known for many parameters.</td>
<td>Previous predictions were later validated by mission data.</td>
<td>Predictable process.</td>
<td>Advanced degree or extensive M&amp;S experience, and recommended practice knowledge.</td>
</tr>
<tr>
<td>2</td>
<td>Unit and regression testing of key features.</td>
<td>Results agree with experimental data or other M&amp;S on unit problems.</td>
<td>Input data traceable to formal documentation.</td>
<td>Deterministic analysis or expert opinion.</td>
<td>Sensitivity known for a few parameters.</td>
<td>Used before for critical decisions.</td>
<td>Established process.</td>
<td>Formal M&amp;S training and experience, and recommended practice training.</td>
</tr>
<tr>
<td>0</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
<td>Insufficient evidence</td>
</tr>
</tbody>
</table>

Table 2.1: NASAs CAS, presented in Ref. [35]
3 Standards for model and simulation integration

As the spread of the M&S applications increased, the ability to exchange models from different domains or tools became necessary. To better understand this challenge, an example will be provided. Figure 3.1 illustrates a simulator that consists of three different simulation models: a flight dynamics model, developed in FORTRAN; a model for the servo actuators of the aircraft, in this example modelled in Simulink®; and a model for the hydraulics supply system of the aircraft, in this example modelled in Dymola®. These three tools are incompatible with each other, and the three simulation models, in their original form, cannot be simulated together; however, information exchange between the models is required for a complete aircraft simulation. For instance, the flight dynamics model requires information about the servo positions, which are dependent, among other things, on the pressure in the hydraulic system. The models then need to be “translated” into a common language. To avoid rewriting models into other languages, several standards for model and simulation integration have been developed. For instance, DAVE-ML, developed by NASA ¹, is tailored specifically for the exchange of flight dynamics simulation models [29], and the IEEEs² High Level Architecture (HLA) standard specifies a framework for the description and structuring of a simulation [25, 37, 38]. The framework developed in the AGILE project [39] developed a collaboration workflow involving a suite of tools and a common aircraft data definition [40], thus enabling the exchange of results and easing the interdisciplinary collaboration. Paper I provides more information on the current SotA on this topic. The work in this thesis is performed with the help of the FMI and the SSP standards, so these will be described in more detail.

¹NASA: The National Aeronautics and Space Administration, https://www.nasa.gov
²IEEE: Institute of Electrical and Electronics Engineers, https://www.ieee.org
3. Standards for Model and Simulation Integration

Figure 3.1: An example of interconnected system simulation models developed in tools originally incompatible with each other

3.1 The Functional Mock-up Interface standard

The FMI standard is an open standard developed by the Modelica Association [26]. Since its release it has become the most used standard for system simulation model exchange and co-simulation. At the time of writing, more than 170 tools support this standard. The standard defines a way of exchanging simulation models via EXtensible Markup Language (XML) files, C-code and binaries, embedded in a ZIP file and an Application Programming Interface (API).

When a simulation model is exported with the help of the FMI standard, a binary file is created - the so-called Functional Mock-up Unit (FMU). An XML file is also created - the Model Description file; this file defines the model interface and its properties. Depending on the type of interface, there can be three types of FMUs:

1. FMUs for Co-Simulation: the FMU contains its own solver;
2. FMUs for Model Exchange: no solver included in the FMU; the importer is tasked with the numerical integration; and
3. FMUs for Scheduled Execution: the importer triggers the execution of model partitions.

The importer, sometimes called the Master Simulation Tool, is the tool executing the FMUs, with or without other models.

Only FMUs for Model Exchange and Co-Simulation were used in this thesis. The Model Exchange part of the standard provides an interface for solving hybrid Ordinary Differential Equations (ODEs); that is, ODEs that include events - discrete points in time where a discontinuity occurs; for instance, a Boolean variable turning from true to false when a condition is fulfilled. As mentioned before, the solver is
3.2. The System Structure and Parametrisation standard

not included in the Model Exchange FMUs; only the model algorithm is provided. The solver is provided by the importer, and it is responsible for all simulation-related activities, like advancing the simulation time, exchanging data, numerical integration, handling events and so on.

In a Co-Simulation FMU the solver is incorporated together with the model algorithm. Data exchange between FMUs is executed at discrete time communication points, which fall within the responsibilities of the co-simulation algorithm in the importer. Between the communication points, the models within the FMUs are solved with their internal solvers. The co-simulation algorithm is also responsible for advancing the overall simulation time, data exchange event handling and so on.

3.2 The System Structure and Parametrisation standard

While the FMI standard is related to the exchange of individual simulation models, the SSP standard [27] is concerned with the exchange of coupled simulation models, for instance several FMUs, covering their architecture and parametrisation. The standard comprises several XML file formats that define the architecture of the exchanged simulators, their signal flow and their parametrisation. Two of these types of files are used in the presented research:

1. the System Structure Description (SSD) format: it describes the hierarchy of systems and components; and
2. the System Structure Values (SSV) format: it provides values for the parameters relevant for a component or system.

Both these files are packaged in a ZIP file, together with other referenced models or resources, that could include other SSP files or FMUs.

When a system simulation model is FMI-compatible, its interface abides to the standard format. The input and output parameters are easy to identify, and the inner workings of the FMUs may be either visible or completely hidden.

3.3 Benefits of using open standards

The practical implications of working with FMUs and SSPs are varied and far-reaching. One of the biggest benefits is the ease of model exchange (with or without co-simulation properties) between the tools compliant with the standards. Not only does this reduce the integration efforts required to simulate models stemming from different applications, saving valuable man-hours during the development process, but it also enables the simulation of systems that, perhaps, would otherwise have been unfeasible. Moreover, the FMUs can be generated either as white boxes or as black-boxes, masking the inner workings of the models contained. This enables
3. Standards for Model and Simulation Integration

easier collaboration with suppliers, clients or even between different legal entities of the same company. However, although interfacing standards greatly facilitate model reuse, it is important to note that the models need to match not only in form, but also in function. In this respect, although the simulation model properties from Chapter 2 are always important, they are even more so when interconnecting models from different disciplines, with or without visible equation layers.
As stated before, during the initial phases of a typical aircraft development project, several departments work on their respective system concepts, developing system simulation models to fit their simulation needs. This means, in practice, that several models of the same system often simultaneously exist scattered throughout the company. An example of this is the simulation model of a hydraulic supply system, which is required at the department concerned with its development and those departments concerned with the development of the aircraft servos.

The reason for this is the multidisciplinary nature of developing an aircraft. One single system interfaces many others, resulting in the information about that system being required in many places. Although the domain separation between the engineering disciplines is well established, the information content of the simulation models they create is less so. This is especially clear during the concept development phase, where there are many iterations of the Overall Design that imply significant changes to the aircraft subsystems.

Figure 4.1 presents just a few of the disciplines involved during aircraft development. At the conceptual stage, Overall Design needs to take all of the disciplines into account; however, the development part of each discipline occurs at the responsible department. Changes to any of the systems propagate to the Overall Design, and, consequently, affect the other systems. Figure 4.2 illustrates the principle of change propagation. It could be said that it is a combination of both top-down and bottom-up design approaches, where the design changes are first broadcasted from Overall Design to the systems developers and then the system design updates are
4. REUSE OF SYSTEM SIMULATION MODELS IN AIRCRAFT CONCEPTUAL DESIGN

Figure 4.1: Some of the disciplines involved in the conceptual design of an aircraft. Overall Design takes into account factors from all disciplines, but the main part of the development is performed within the disciplines themselves.

communicated back to Overall Design. Naturally, informal discussions between all the involved parties are also a means of information dissemination.

Figure 4.2: Change propagation between Overall Design and system design.

A proposed enhancement to this way of working would be the possibility for systems engineers (or domain experts) to independently evaluate the consequences of the modifications of their own systems. In practice this means having access to all of the system simulation models of the other domains the system has influence on. As mentioned in Chapter 2, given that the system simulation models are developed in the tools best fit for purpose, these models might not be compatible with each other. This is where interfacing standards from Chapter 3 come in.
4.1 The conventional Aircraft Conceptual Design process

To understand the benefit of this additional step in the way of working, and how it can be implemented already at the ACD stage, a brief overview of the traditional ACD methods is given. The method is presented by Raymer in [41]; others [42] work just as well. Then, a description of the ACD framework that is the backbone of this thesis is presented.

4.1 The conventional Aircraft Conceptual Design process

The most important steps in the ACD process, according to Raymer [41], are shown in Figure 4.3. Mathematical models exist for each of the involved disciplines - aerodynamics, weights, propulsion, structures, landing gear, radar signature, sensors, and many others. The equations they are based on are static, and the parameter values are based on empirical relations.

The first sizing iteration is based on rough calculations of the desired aircraft’s aerodynamics, weights and propulsion. When a rough design that meets the requirements is set (the “initial layout”), more thorough calculations are performed. At this stage trade-studies can be executed in order to discover a satisfactory solution to the given requirements. Although the models employed during this step are more detailed than the rough estimations from the initial sizing, they are still based on static equations and approximations based on empirical data. Therefore, the aircraft can only undergo steady-state performance evaluations.

![Figure 4.3: The ACD process, from [41]](image-url)
4. REUSE OF SYSTEM SIMULATION MODELS IN AIRCRAFT CONCEPTUAL DESIGN

4.2 The proposed Aircraft Conceptual Design framework

The sizing methods described by Raymer [41] have been implemented in a Modelica library in Paper II and further improved in Paper III. At the time of writing, the library, from now on called the Aircraft Conceptual Design (ACD) library, contains models for the geometry of the aircraft, its aerodynamics, propulsion, atmosphere and for a standard mission. The geometry and the aerodynamics simulation models are built so that they automatically generate a new wing design (in terms of wingspan) based on some given input parameters. However, the algebraic relations can be reordered and/or rewritten such that other geometrical parameters become a design variable should that be required. The original version of the library was developed by Franzén and Hällqvist for the design of a transport aircraft [43]. In the original version, however, the geometrical properties of the aircraft were calculated in other software and manually imported as parameters in the Modelica library.

Paper II introduces an optimisation framework, based on OMSimulator [44] and its Python API. The framework uses the Complex-RF [45] or the Simplex [46] optimisation algorithms. The framework is application-agnostic, meaning that it can be used for any optimisation task. In Paper II and Paper III it is used to solve the fictitious task of designing a fighter - finding the appropriate engine size and the wing span required to fulfil a certain flight mission. The system simulation models are provided by the ACD library in the shape of FMUs. The aircraft sizing task reduces to the import of desired FMUs in the optimisation framework, defining an appropriate cost function and allowing the optimisation process to run its course. A schematics of the sizing optimisation framework is presented in Figure 4.4, reflecting the use case from Papers II and III.

Figure 4.4: Schematic view of the proposed ACD framework, used in Papers II and III to generate an aircraft design to fulfil a predefined mission. The subsystems 1 to N represent any system simulation model to be included in the sizing process.
4.3 Legacy model integration

So far it might not seem that this library and accompanying optimisation framework is too different than already existing ACD suites. The fundamental equations are, naturally, the same, as the governing physical processes are the same irrespective of the modelling language. However, there are three details that set this framework apart. A first distinction is the added benefit of using the FMI and SSP standard compliant software. The advantages of this solution have already been discussed in Chapter 3. A second benefit is given by the fact that the library is written in Modelica, which is a freely available, object-oriented language for modelling large, complex, and heterogeneous systems [47] that is especially suited for dynamic system simulation models. Therefore, as a proof-of-concept, in the Modelica library simple dynamics were included in forms of accelerations along the x-axis. This allows for performance evaluations, not only in the typical steady-state flight conditions, but also during dynamic aircraft manoeuvres. The third benefit is that the framework can generate an entire aircraft design based on some given inputs. It could be used by any department desiring to investigate the impact of their own system modifications on the entire aircraft, or the impact of modifications to the entire aircraft on their own system. It removes the necessity of installing, maintaining, learning and understanding different tools required for the simulation of different systems when the desired output is a rough design or a trade study. The level of fidelity of the simulation models in the Modelica library is low, but they can still provide valuable information when used as intended, especially in the initial phases of a new aircraft concept.

The most important benefit, however, is given by a combination of the properties from above. The ability to integrate any FMU in the framework and to simulate dynamic models early in the PDP opens up for the possibility of taking advantage of the information contained in the higher fidelity system simulation models that already exist at the aircraft company right from the start, provided their properties (intended use, validity, credibility and level of fidelity) are appropriate for the task to be solved. The idea of including higher fidelity simulation models early in the conceptual design stage is not new. It has already been applied in academic research projects on the conceptual design of aircraft, through the use of variable fidelity aerodynamics models coupled to flight dynamics analysis [48, 49]. However, what the proposed framework here adds is a more generic solution based on the benefits of the open standards FMI, FMU and SSP, focusing on reusing the legacy system simulation models already existing at the company.

4.3 Legacy model integration

In Paper III the ACD framework is tested by including industry-grade legacy models in the optimisation loop. These models are originally developed for a Saab Aeronautics in-house simulation environment called Aircraft Rigid body Engineering Simulation (ARES), which currently does not support the FMI standard. The
4. REUSE OF SYSTEM SIMULATION MODELS IN AIRCRAFT CONCEPTUAL DESIGN

Figure 4.5: Radar plot of the credibility levels for the handbook method-based model and the legacy model

system simulation models are written as FORTRAN procedures and share the same function call structure. By using the Modelica possibilities of integrating FORTRAN code as external functions [47], an integration wrapper was developed. Given that the ARES models have identical function calls, the wrapper works with every FORTRAN model from ARES. After integration in the Modelica code, the ARES models are then exported as FMUs and included in the framework.

To evaluate the effect of including legacy models already at the ACD stage, a sizing task is performed first with simulation models from the Modelica ACD library, similar to the one executed in Paper II. The same task is then performed with a legacy aerodynamics model from ARES. As expected, the resulting aircraft differ in outer geometries and engine size, as the two aerodynamics models provide different values for the aerodynamic coefficients.

The difference in the optimisation results sparks the question of which simulation model is closer to reality. As this is impossible to tell at the conceptual design stage, since the aircraft is not yet built, the answer lies in the degree of confidence one has in the simulation models, or, as mentioned in Chapter 2, the credibility of the model. The CAS presented in Chapter 2 is a good starting point for such an assessment. The exercise of evaluating the credibility of the traditional ACD methods for the double-delta aircraft design from Paper III can be made. The results are presented in Figure 4.5 (the models that resulted from the traditional ACD methods are here called handbook models). The figure shows that, all other things equal, reusing legacy models when suitable increases the credibility of the simulation results. The credibility assessment of each of the eight criteria is a personal interpretation, based on the following:

- Verification:
4.3. Legacy model integration

- handbook models: 1. The physics and the empirical relations in the models are verified, but there are no formal unit and regression testing (or similar) activities on the developed models
- legacy models: idem above

• Validation:
- handbook models: 1. The conceptual models agree with referrent data, but there are no experimental values to compare to.
- legacy models: 2. Results agree with other M&S performed on similar problems

• Input Pedigree
- handbook models: 2. The input data comes from formal requirements.
- legacy models: idem above.

• Results Uncertainty
- handbook models: 1. Qualitative estimates. No deterministic analysis performed
- legacy models: 2. Some deterministic analysis and expert opinion.

• Results Robustness
- handbook models: 1. Only qualitative estimates and no formal sensitivity analysis performed.\(^1\)
- legacy models: 4. As the number of input parameters is limited, and parameter sweeps lie as a basis for the tabulated data all dependencies are mapped.

• Use History
- handbook models: 1. The model passes simple tests, but it has not been used before for critical decisions related to a double-delta wing configuration.
- legacy models: 4. CFD models are the de-facto standard for aerodynamics.

• M&S Management \(^2\)
- handbook models: 4. Continual process improvement

\(^1\)Sensitivity analysis could be performed and the value could be increased up to 4, depending on the time budget.
\(^2\)For the use-case of this thesis, for both M&S Management and People Qualifications the handbook model and the legacy model have the same value as it is the same people that work with the models and the same process that is applied.
4. REUSE OF SYSTEM SIMULATION MODELS IN AIRCRAFT CONCEPTUAL DESIGN

- legacy models: idem above.

- People Qualifications
  - handbook models: Extensive experience in and use of recommended practices for this particular M&S.
  - legacy models: Extensive experience in and use of recommended practices for this particular M&S.

It should be noted that the legacy models used here are from a project that has not completed the testing phase, and therefore the Verification and Validation scores are very conservative. If the simulation models would have come from projects that have undergone the complete V&V phase, the scores would be higher. However, even in the current state there is a clear benefit of reusing legacy models.

4.4 Flowchart for simulation model creation

Given all the information presented in the previous section, it follows that there are cases when legacy models would be more appropriate to be used for new projects, or when creating a new model (regardless of whether it is based on handbook methods) might be more appropriate. The two paths are summarised in Figure 4.6. Note that other decision points might apply depending on the application area of the simulation models, the nature of the project, the structure of the company and so on.

If a new simulation model needs to be created, then appropriate model development methodologies (such as the ones identified in Paper I) can be applied. If a legacy model can be reused, and it cannot be easily integrated into the simulation environment, then the following four steps can be taken (Paper III offers more details on these steps - a fifth one is included there at the beginning of the process, but it is covered by Figure 4.6 - and their effect on the integration process as a whole).

1. Model interface development to ensure use-case compliance and, if native tool FMI support is unavailable, model export according to the FMI standard
2. Integration testing of the exported FMU
3. Model verification to identify any numerical errors. If such errors are present, model or solver adjustments may be needed to mitigate the problems
4. Once exported and verified, the legacy model is ready to be used
Figure 4.6: Workflow describing the decision points when choosing between reusing a legacy model or creating a new one.
Discussion and Conclusion

5.1 Overview of the appended papers

There are three papers appended to this thesis. Paper I lays the foundation for this work. It is the summary of a literature review that aimed at identifying the SotA in flight simulation. To broaden the search area, simulation models have been regarded as "risk-averse scientific software" [16]. The key findings are that the development processes that are traditional to the software industry are not entirely applicable to the domains where the main product is knowledge, since the extent of the problem to be solved is defined as the solution is being developed. The author considers the M&S efforts in the aeronautics industry as being such a domain. Second, the study identified that efforts were being made to increase model reuse and interdisciplinary collaboration through the development of several standards and collaboration workflows; the FMI and SSP standards being some of them.

Paper II develops a simulation framework for ACD by implementing an aerodynamics model based on Raymer’s handbook methods [41] in the Modelica language. The model serves as the basis for automatically sizing a generic fighter aircraft. The aerodynamics model is added to a pre-existing Modelica library for aircraft sizing that included models for the flight mission, atmosphere and aircraft propulsion. The models are then exported as FMUs and integrated into an optimisation framework developed in Python. The novelty lies in the creation of a tool-agnostic workflow for ACD that allows for the inclusion of transient phenomena. This is done through the integration of dynamic simulation models of aircraft subsystems using industry-wide
5. DISCUSSION AND CONCLUSION

Paper III further develops the framework developed in Paper II by enabling the inclusion of legacy simulation models originally written in FORTRAN for the ARES simulation environment. These legacy models are imported in a Modelica-based tool with the use of a wrapper, and are exported as an FMU. The FMUs of the legacy simulation models are integrated into the optimisation framework in the same manner as the other system FMUs. There is scarce information about the integration of FORTRAN models into Modelica code available online, although the Modelica specification allows for it [47, 50]. As the FORTRAN legacy code-base is extensive in the field of aeronautics (and not only), the existence of a relatively simple integration solution for it in modern simulation software can extend its lifetime and relieve companies of the burden of having legacy simulation models (or software) running on deprecated machines that cannot be replaced in case (or when) they fail.

Paper III introduces a five-step integration method for legacy simulation models. The first step in the method, the assessment of simulation model suitability to the target system, is further developed in this thesis in Figure 4.6. This enables the reuse of legacy models already at the conceptual stage, partly relieving the systems engineer of the V&V activities that have already been performed for that particular legacy model. Moreover, there is more confidence in the simulation result, since the legacy simulation model’s level of credibility is generally higher than for a low fidelity handbook model. Additionally, reusing legacy system simulation models allows for a more thorough design space exploration already at the conceptual stage. The inclusion of higher fidelity simulation models might highlight secondary effects and interactions between the systems that cannot be captured with low fidelity models due to simplifications.

5.2 Discussion

This thesis stems from an industrial use case in the aeronautical domain, but it is likely that its findings can be valid in any field that applies M&S in their PDP, namely the nuclear industry, weather prediction applications, automotive industry, processing industries, manufacturing industries, etc. Legacy model reuse might be of interest even for academia, where the developed models and software usually outlive the employments of the researchers who have originally created them.

To begin with, the choice of research method should be addressed. *Industry-as-a-laboratory* [6, 7] proves to be the correct approach, as it enables taking industrial real-life problems and expand them through academic research. Concrete problems, that at a first glance might seem as an implementation issue rather than a research one (for instance the ability to simulate various vehicles with minimum overhead in terms of simulation resources and modifications to the simulation models), turn
5.2. Discussion

into research questions as soon as the academic approach is taken. The SotA review of Paper I identified that the problems encountered during M&S for aeronautics applications were not merely "implementation" issues, but systemic shortcomings in industry and, sometimes, academia. As such, the identified research front could be used as a starting platform for the subsequent work.

As mentioned in the introduction, this work places itself at the intersection of four domains: aeronautics, ACD, vehicle systems, and software engineering. Naturally, performing extensive research into all four domains falls outside the scope of this thesis; however, the process of modelling is related to all of them, and thus benefits from their insights. The core idea of the work is model reuse, and it impacts the four domains in different ways. Through the model properties from Figure 2.7, the aeronautics and vehicle systems (and other technical disciplines that might be included) are affected. Through the possibility of interconnecting models from different disciplines and tools, and through the reuse of legacy models the aircraft conceptual design domain is affected. Finally, through the methodologies applied when developing models and the approaches taken when integrating them, the field of software engineering is impacted. With all certainty there might be other implications, but these are enough to exemplify the central role reuse has in aircraft simulation.

It was found that the forefront of research lies on two different levels: an overarching one, focusing either on integration of models [25–27] or on integration of tools or developing collaborative frameworks [39]; and a domain-specific one, where developments in domain-specific methods and tools are obtained. This work places itself somewhere between these two levels: it deals with the realisation of a simulation model in terms of the knowledge that is required and how that knowledge is best obtained (through reuse or new development). The work assumes the perspective of domain specialists taking advantage of existing knowledge from their own systems or others'; it does not dig into specific modelling techniques best fit for each domain, nor does it investigate the suitable workflow that would maximise the information (re-)use between the different domains. Nevertheless, collaboration is identified as a key enabler; further research into how it is best achieved in the reality of an industrial setting is required.

When it comes to the practical implications of reusing legacy models already at the conceptual design stage, it should be mentioned that having access to higher fidelity system simulation models this early in the PDP is a double-edged sword: it narrows down the overall design space, as choosing a legacy model steers the design in the path of the legacy system. However, it enables a more thorough design space exploration of the investigated concept, as more interactions between the system and the rest of the product can be studied. Therefore, careful consideration is required when considering these two options, as they have different outcomes.
5. **Discussion and Conclusion**

High fidelity legacy models have a higher information content than handbook models built specifically for the task at hand. In Paper III, although not stated in the text, the integration wrapper was developed first on the use case of the atmosphere model. This is a simple model, with equations that are well known and easy to verify, thus a best case on which to test an integration wrapper. However, the legacy FORTRAN atmospheric model includes several variables that are not relevant to the project at hand, or that are calculated somewhere else in the other domain models (for instance the dynamic pressure or the Mach number). All these variables need to be accounted for when integrating the atmospheric model, generating additional workload for information that is, in practice, not required from that particular simulation model (see the discussion from Chapter 2 on intended-/target-/legacy use). Including calculations that are not necessary might incur simulation execution time delays and generate numerical problems. Moreover, any extra information needs to be handled, as it could also cause confusion when a user requiring the value of a variable is confronted with the situation of choosing between the same variable calculated in two different simulation models. Since these could be in the form of a black-box FMU, it might not be obvious which one to use.

The mere possibility of simulation model reuse already presents benefits, besides the added benefits of actual model reuse. M&S with reuse in mind highlights the importance of thorough documentation and decision tracing. These tasks can be viewed as unnecessary by some, especially in the beginning of a project that has not yet received "the green light", when the connection to formal certification requirements is not yet obvious. However, improper documentation lies as an impediment to model reuse, as identified in Paper I, thus creating a vicious circle where the only escape is new development. Thorough simulation model documentation not only increases the traceability of design decisions, but it also acts as an agent of knowledge dissemination. The simulation models used in aircraft development have long life cycles, but the present-day work environment encourages a more dynamic professional path than what was customary some decades ago. Therefore, a simulation model will continue to live long after the initial model developer has moved on in their career. Chances are that a simulation model that starts as a rough sketch will have a central role later in the development process, and writing the necessary paperwork after the model is already built is less than ideal. Requirements on proper documentation ensure that as much as possible of the knowledge included in the simulation model is transmitted to the developers inheriting it.

Looking at the use of integration standards, there are more benefits than the ones presented in Section 3.3, from where the ease of simulation models and/or information exchange within an organisation and between organisations deserves to be mentioned again, as it has a strong impact on facilitating collaboration. Open standards are also free to use, and, moreover, one can get involved (as an organisation) in their development. The scientific community is an advocate for open-source standards and software, and the progress is fast paced. If in 2019 there were nearly
100 tools using the FMI standard [12], at the time of writing there are more than 170 [26]. Open standards are making their way even in proprietary software, for example Dymola® and MATLAB® [51, 52], so there is no reason to believe that they will become obsolete in the near future. Aside from their wide-spread use, they do not incur any licensing or recurring costs either, so there are no financial impediments to their use within a company. Having a "de-facto standard" to refer to within a company should also increase model reuse awareness.

However, the most important factor in the model reuse equation is neither the existence of a perfect M&S process, nor the implementation of open integration standards. The mindset of the people working in the company must allow for an increased inter-department collaboration. Although it was mentioned before that model reuse can be as easy as "plug-and-play", in reality this will only be the case from a technical perspective. Establishing whether a simulation model has the correct properties for model reuse, and going through the steps of the flowchart in Figure 4.6, requires collaboration efforts, at least between the department requiring a new simulation model, the department responsible for the legacy simulation model and the department fulfilling the Overall Design function. Increased collaboration is key to model reuse, and it must happen from the beginning of the development process. The organisation also needs to adapt to this new mindset, leveraging the long-term perspective of maximising model reuse and increasing overall knowledge in each individual development project.

One should also bear in mind that the ultimate result of a M&S effort is not necessarily the modelled system itself, but the knowledge that is produced. Therefore, in the end, the path towards the "right" design will not be straight even if one makes all the "right" choices from the beginning. With every piece of new information uncovered through M&S, through reuse of legacy simulation models or through building new simulation models from scratch, a new design iteration, be it formal or informal, is initiated, as the newly discovered system properties need to be weighed against the pre-existing information. This is a fundamental prerequisite for obtaining an optimal and robust design; however, exploring all the possible design options requires time. Therefore, all the design generation alternatives and the design exploration possibilities should be weighed against the available time, budget and human resources. One could always find more topics to explore and situations to simulate, but a successful approach is not the one that explores all possibilities; it is the one that explores just enough of them.
5. DISCUSSION AND CONCLUSION

5.3 Conclusion

The Research Questions mentioned at the beginning of this work are restated below and the author believes that she has presented sufficient information to justify the following answers.

**RQ1. How can an aircraft simulator be designed to enable aggregation, integration, and reuse of simulation models from different technological domains, with different levels of fidelity?**

In the context of this work, and for aeronautical and similar applications, a simulator should take into account the following:

- well-established processes for simulation model development that facilitate model reuse and knowledge dissemination;
- the ease of simulation model integration, through open integration standards, such as FMI and SSP; and
- an appropriate project and organisation structure and culture that allows for inter-disciplinary collaboration through simulation model exchange, simulation results exchange and multi-disciplinary system analysis.

The creation of an aircraft simulator starts with the creation of the models included in it. A simulation model development process that properly identifies that the main product when developing a model is knowledge (for instance, Kelly’s knowledge acquisition process [53]), combined with the rigour of a development process adapted to the aeronautics industry [24], should facilitate the development of reusable simulation models.

The second requirement is to make use of modelling and simulation standards. Papers II and III highlighted the benefits of using the FMI and SSP standards when exporting models and simulators. A simulator architecture allowing for a “plug-and-play” type of simulation model connection considerably reduces the model integration efforts, leaving more time for the core activities of the simulation engineers, namely simulation and analysis, enabling a more thorough design space exploration.

The third requirement highlights the impact the people working with the simulation models have on how they are (re-)used. The company structure, the project setup and the work environment should allow for and encourage inter-disciplinary collaboration. Paper I identified several reasons why engineers prefer to develop their own models rather than reuse others, and some of the reasons could be categorised into not being offered the correct premises: a lack of appropriate funding for correct model development, a lack of education into correct work practices, an inability to deal
5.4. Future work

with the changing nature of the project requirements and so on. Therefore, the focus should lie not only on the technological aspect of designing a versatile simulator, but also on the development processes related to it so that its benefits can be maximised. This includes continuous education for all the workforce involved in the processes, management and tools existing at the company, as well as both a specialisation and broadening of the engineering competences of everyone involved.

RQ2. What are the challenges encountered when reusing legacy system simulation models in new development projects?

The flowchart in Figure 4.6 presents the challenges to be overcome when deciding whether to reuse a simulation model. In order to make the correct decision at each decision point (represented by the rhombus in the flowchart), several requirements need to be satisfied. The DP1 assumes that an appropriate storage of the simulation model exists (whose exact form is dependent on the company). If no one knows about the legacy model, it cannot be reused, irrespective of how well it fits the requirements. At DP2, DP3 and DP4 the necessary documentation needs to be present to be able to answer the respective questions. DP1 through 4 set requirements on the structure of the company and its M&S management processes. From DP5a and DP5b and onward, the answers are more dependent on the simulation engineer(s). Depending on their knowledge of the simulation environment and on the available support for the tasks outside their own technical domain, the answer to DP5a, DP5b and DP6 can be both yes or no, with neither being better than the other. DP5 and DP6 depend on the entire context of the work, human factor included, whereas DP1-4 are objective in nature.

Probably the most challenging problem is to implement an appropriate organisation structure that actively encourages increased collaboration, model reuse and knowledge dissemination. This is especially difficult to overcome when each development project has its own budget with no room for extra costs. Organisations might mitigate this by funding technology/methods/tools improvement projects on the side of their product development ones, but as long as the results are not integrated into the day-to-day work there is little to be gained. The author does not possess the answer to how to do this, and there is probably no answer that will fit all organisations; however, the key to success lies in convincing the entire organisation, from the leadership to the engineers directly involved in the projects, of the benefit of thinking long-term even on short-term assignments.

5.4 Future work

The proposed work is to be seen as a "proof-of-concept". As such, the Modelica-developed simulation models include simple dynamics in the flight equations. It would be beneficial to include dynamic simulation models of a higher level of fidelity even for the vehicle systems or propulsion. Performing a sizing task with
subsystem dynamics and the inclusion of some subsystem parameters in the cost function of the optimisation routine would shed more light on the gains of having higher-fidelity simulation models this early in the aircraft development process. Moreover, the resulting aircraft geometries when using the developed Modelica library have undergone plausibility checks with respect the sought geometries. The library has not been subjected to detailed V&V activities; these are recommended if one is interested in the generation of more accurate designs with it.

Furthermore, since the start of this work a 6-degrees of freedom library for aircraft simulation has been made available in Modelica [54]. This could be integrated into the framework, thus expanding its use area to flight dynamics analysis. Together with the dynamic system simulation models mentioned above, the basis is set for a truly flexible simulation framework, which can be tailored to specific domain analyses right from the get-go of a project, with minimum overhead.

Further future efforts include performing a continuous analysis of the scientific SotA. For instance, during the elapsed time since the publication of Paper I, a new standard for simulation integration has appeared [28]. A continuous supervision of the SotA during the entire PDP should lie as a core activity for any development process.

The work presented in this thesis has been mainly performed by a single person at a single company and in a single research environment. However, valuable insights into the PDP and ACD process was gained through collaborations and interactions with other departments at both Saab Aeronautics and at Linköping University. A possible development path would be to use the proposed framework at other departments within the company and even in other industrial settings.
Bibliography


Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

https://doi.org/10.3384/9789179294762