The Aircraft and its Manufacturing System: From Early Requirements to Global Design

Anouck Chan & Anthony Fernandes Pires
Thomas Polacsek & Stéphanie Roussel*

Abstract

The design of an aircraft manufacturing system depends highly on the design of the aircraft itself. In this work, we propose an approach based on conceptual modelling and optimization methods that allows to take into account the impact of an aircraft design on its assembly line design. We start by eliciting early requirements in a real industrial use case context. Using Goal Oriented Requirements Engineering, we highlight the dependencies between the systems as well as the key elements that must be optimized to obtain an optimal global system. Then, based on a conceptual model and operational research, we present the tool that we have developed to support the development of an optimal overall system. We analyse the experiments realized on different aircraft designs, and we identify and summarize the lessons learned from this experience.

1. Introduction

In industry, there are systems that are designed in an asynchronous way and that are dependent on each other. Such systems can be found, for example, in the context of software development and operations, where operational environment system is based on software design. In the embedded systems context, hardware and software designs can also sometimes be out of synchronization but depend on each other. If hardware is decided first, choices made on it strongly impact software (*e.g.* because of memory limitations).

In this paper, we focus on another example of such systems, namely an aircraft and its manufacturing system. Indeed, aircraft development is not only limited to the elaboration of an aircraft design, but also includes means to build it. For each new aircraft family, a dedicated industrial system must be developed. Even if it is built upon existing means like basic plants, new assembly lines (along with new tools, robots, *etc.*) must be developed to cope with the specificity of each new aircraft family and with production objectives (*e.g.* number of aircraft to produce, rate). An aircraft and its manufacturing system are strongly linked to each other. Every time the design of the first changes, it may be necessary to redesign the latter. In addition, it is impossible to design the manufacturing system until certain aircraft design choices are made. Lastly, the performance of the manufacturing system highly depends on the aircraft design.

For designing these two systems, it is essential to anticipate as soon as possible the impacts of aircraft design choices on its manufacturing system. For instance, it could be possible that a manufacturing requirement such as *reach a production rate of 50 aircraft per month* is not satisfied because of the aircraft specifications (*e.g.* the chosen material which requires a complex process for drilling and junctioning). In this example, the aircraft design must be modified, which is quite heavy. In fact, it requires to make new high-level changes (*e.g.* material choice), cascade them and then assess if this new design is consistent with the manufacturing requirement. Being able to

*Authors version, The Aircraft and its Manufacturing System: From Early Requirements to Global Design assess, even partially, the impact of the aircraft design on the manufacturing goals would allow earlier trade-offs in case of conflict and thus a more satisfactory overall specification. It would also prevent high costs due to late design modifications. A good design for the two systems is more a matter of global optimum than of the separate optimization of each of them.

In this paper, we introduce an approach that combines proven modelling techniques and optimization methods to support the design of the aircraft and its manufacturing systems. The aim of this approach is to provide methods and tools for taking into account, from the early design phases, the manufacturing performance in the design of the aircraft.

This paper is organized as follows: in Section 2, we describe our industrial case study. Section 3 and 4 are respectively dedicated to goals elicitation and to overall optimization and provide details about the application on the case study. Then, we discuss lessons learned in Section 5. Section 6 is dedicated to related work and we conclude in Section 7.

2. Industrial Case Study

In this section, we detail the main features of the industrial case study we have worked on. The design of the aircraft has an impact on many aspects of manufacturing : the workplace ergonomics, the space available to perform assembly operations, the need for very specific tools and machines, the means of transporting the aircraft components according to their size, *etc*. Moreover, the way the aircraft is broken down into parts can induce more or less assembly operations. For instance, installing several light parts could be easier than a large heavy one. However, it could require much more connection operations. Therefore, the objective of reaching a high industrial efficiency is also linked to the aircraft design choices.

Our industrial study framework deals with the design and manufacture of a specific component: the structure of the aircraft's fuselage, namely the *airframe*, and its assembly line. The fuselage is designed to provide spaces for the crew, passengers and cargo and to withstand the aerodynamic forces. It is also the link between the wings, the tail and the nose. The airframe is composed of a keel beam, circular frames, linear spars and skin panels which must be assembled. An assembly line is a sequence of workstations that corresponds to all the assembly stages of a product. In the aeronautics industry, assembly lines can be found at both Boeing [17] and Airbus [16, 24].

In this case study, we have worked with two *aircraft architects*, who design the fuselage, and one *assembly architect*, who designs assembly lines. All three of them have been working for a major aircraft manufacturer for many years. The aircraft architects are specialised in architectural design, *i.e.* making choices and designing the elements and systems of the aircraft without going into details. For instance, they do not consider the number or the position of each bolt. The assembly architect works on the assembly line organisation and the new assembly tools definition. In this project, he was in charge of the design of a new assembly line for the airframe production. An important point is that the two aircraft architects work in a different department and on a different site from the assembly architect. This is a real silo situation. This project was an opportunity to have a more integrated vision and to take into account the impact of the product on the production. Through five half-day working sessions over a period of three months, we supported the architects to clarify the goals, the constraints and what was not in the scope of this case study. To this end, we proposed a *Goal-Oriented Requirements Engineering* (GORE) model and briefly trained architects in this approach. Then, for three months, at the frequency of a meeting every week with the architects, we worked on the problem formalisation and on the design of the tool presented in the following sections. The architects provided us with all the information regarding the assembly operations related to the different aircraft designs and the assembly methods and tools. From our side, we helped them to clarify their objectives and we provided them with a tool to evaluate the impacts of the product on the production.

In our use case, the problem was to design an aircraft fuselage as manufacturable as possible, *i.e.* as easy and as fast as possible to assemble and with a not too expensive factory. More precisely, architects want to be able to evaluate aircraft design solutions at early stages of the design process and, to do this, assess them with respect to the industrial performance of their corresponding assembly lines. Note that the concept of industrial performance was not clearly defined at the beginning of our study.

In the two following sections, we describe how we tackled this problem. Starting from requirements elicitation, we explain the relevant metrics we have identified to measure the assembly line performances. Then, we present a tool that allows to perform numerical comparison of two fuselage airframe designs.

3. Identify Goals and Objectives

The first step of our work was to understand the dependencies between the aircraft and its manufacturing system and specifically how the design choices impact assembly line performance. Expressing such dependencies is not an easy task, as aircraft design and manufacturing design are two different domains. In this section, we show how GORE modelling techniques allow to elicit and understand the goals of each system and their dependencies.

3.1. Model the Goal Oriented Requirements

GORE approaches focus on finely characterizing the interactions among goals of the system and interactions among goals and some other elements. These elements can be internal or external to this system, like actors or resources [29, 6]. Most of GORE frameworks identify two types of goals, namely *goals* and *soft goals*. Soft goals are objectives for which *no clear-cut criteria* indicating whether they are fulfilled can be expressed [21, 13]. Such a distinction is used in frameworks such as *Non-Functional Requirements* (NFR) [20, 6] or i^* [30, 7]. In our context, this distinction is crucial. It allows us to distinguish between what is a non-negotiable constraint and what is flexible. Therefore, we consider that a goal is a constraint, a non-negotiable mandatory objective that a system must satisfy. In reverse, a soft goal is a possible trade-off point, its satisfaction being negotiable. It represents an element of negotiation between the aircraft and the manufacturing systems.

As expressed earlier, the manufacturing system design depends on the aircraft design. Because the designs of these systems are asynchronous, this dependency is a strong one, *i.e.* the manufacturing system cannot exist without the aircraft. This corresponds to the i^* notion of vulnerability of the manufacturing system. This vulnerability is materialised through *dependency relationships*. In i^*, the dependency relation is composed of three elements. The first two elements can be goals or tasks and belong to two different actors and the third is the dependence object (the dependum). It is a unidirectional relationship that indicates a dependency of one actor on another. For instance, the choice of a robot (the depender) depends on the assembly operation to be performed (the dependum), and this operation is defined during the aircraft design (the dependee). Because the dependency relationship is at the heart of i^* framework, we chose i^* for the modelling of our use case goals.

3.2. Application

In our case study, aircraft and assembly architects were not familiar with GORE approaches. None of the goals, soft goals, tasks and dependencies were clearly identified. It took many iterations with the three architects to elicit them and to reach the exploitable goals model given in Figure 1.

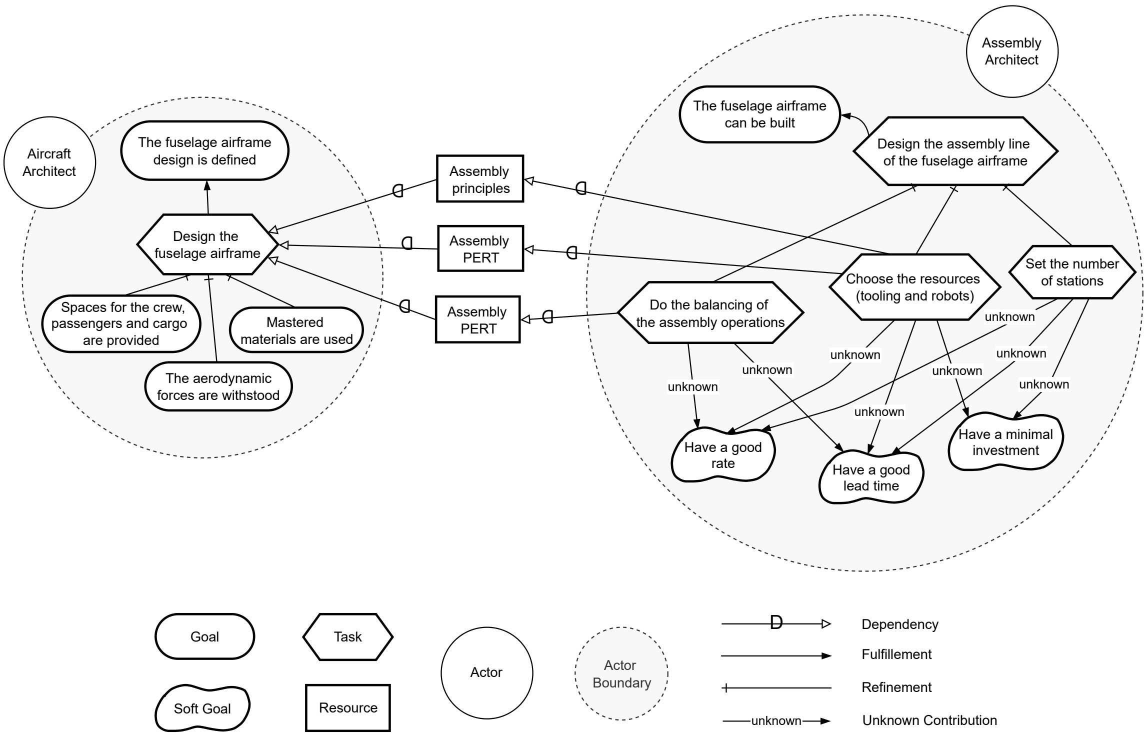

Figure 1: *i* Strategic Rationale model for the design of the fuselage airframe and its assembly line (Model in i* 1.0)*

In this goals model, the task *Design the fuselage airframe* is the main activity of the Aircraft Architect. This task is the mean to satisfy the main goal *The fuselage airframe design is defined* and it is decomposed into three goals that qualify the task: *Spaces for the crew, passengers and cargo are provided*, *The aerodynamic forces are withstood* and *Mastered materials are used*.

For the design of the assembly line, the Assembly Architect main task is to *Design the assembly line of the fuselage airframe*, answering the main goal *The fuselage airframe can be built*. This main task is decomposed into three sub-tasks: *Do the balancing of the assembly operations* which corresponds to the scheduling of the assembly operations based on the *Assembly PERT* (Program Evaluation and Review Technique); *Choose the resources* corresponds to the choice of the resources to perform the assembly depending on the *Assembly principles* and the *Assembly PERT*; and *Set the number of stations* corresponds to the choice of the number of stations composing the assembly line.

Assembly PERT and *Assembly principles* are information produced by the Aircraft Architect. The *Assembly PERT* describes the precedence relationship among assembly operations and their duration. The *Assembly principles* describe how to join the skin panels together and how to fix all the elements together (for instance with joints, fasteners or welding).

In addition, the Assembly Architect has three soft goals that altogether represent the industrial performance of the manufacturing system. *Have a good rate* indicates that the factory must produce as many aircraft as possible. This soft goal is limited by the investments that are made to build the factory, which is expressed by the second soft goal *Have a minimal investment*. The third soft goal, *Have a good lead time*, asserts that the duration to build a single aircraft must be acceptable. They are soft goals because their acceptance is subject to interpretation.

Usually in i^*, tasks impact soft goals. There are various impact types: *make*, *help*, *hurt* or *break*. For trade-offs, tasks can be refined into aircraft design alternatives and assembly line design alternatives. Each alternative impacts goals positively or negatively. From there, many works use these links and propose propagation rules to evaluate the alternatives [11, 3]. In our case, and this is a key consideration, assessing these impacts within the model is impossible: relationships among tasks and soft goals are not monotonic. For example, let us consider the task *Choose the resources*. On the one hand, choosing cutting edge technology resources can help the goal *Have a good lead time*, but might hurt *Have a minimal investment*. On the other hand, choosing cheaper resources that are less efficient can help to *Have a minimal investment* but may hurt the goal *Have a good lead time*. Moreover, refining the different alternatives does not give us more information. It is not obvious that having 6 stations allows to have a better rate than with 5. Even regarding the investment, stations number is not the only cost factor, there are also the tools and machines.

Therefore, due to the combinatory explosion, we are faced with the impossibility of expressing all the alternatives and to manually assess their impacts on goals. So, we choose to model the contribution links to the soft goals as *unknown* and, unlike other works, to perform trade-offs assessment not in i^*, but by using automatic calculus outside of the model. We even go further: besides assessment, we automatically find the best assembly line alternatives for a given aircraft design. To do this, we need to elicit all elements and their relationships that are involved in the satisfaction of the soft goals.

Note that the obtained model looks rather simple. In fact, our problem is to take into account the manufacturing performance for an airframe design. So we do not need to consider all goals involved in the airframe design or in the industrial system design but only the relevant ones. Moreover, we focus on early design phases. Therefore, a refined version of goals is pointless as they would result in the alternatives described previously. Finally, it took several collaborative working sessions to converge towards the goal model presented here. This highlights that all the elements (goals and soft goals, associated tasks and dependencies) were not clear at the beginning of the study.

4. Support the Optimal Overall System Design

In this section, we first present several considerations that we had to take into account in our approach. Then, we describe the methods and tools developed to support the development of an optimal overall system. They are based on a conceptual model and operational research optimization methods.

4.1. Support the Design of an Optimal Solution

Our aim is to support the design of the global system composed by the aircraft fuselage and the assembly line. To do this, it is necessary to have an impact assessment of the fuselage airframe design choices on the assembly line. In order to be useful, this assessment should satisfy four properties. Firstly, the method used to perform the assessment should be clear and accepted by the teams involved in the project development. Secondly, the time required to perform the assessment should be short, in order to allow the teams to quickly have feedback on the current project development when they need one. Thirdly, it should not be subjective, *i.e.* its value should be the same regardless of who performs it. In order to be objective, metrics can be used along with

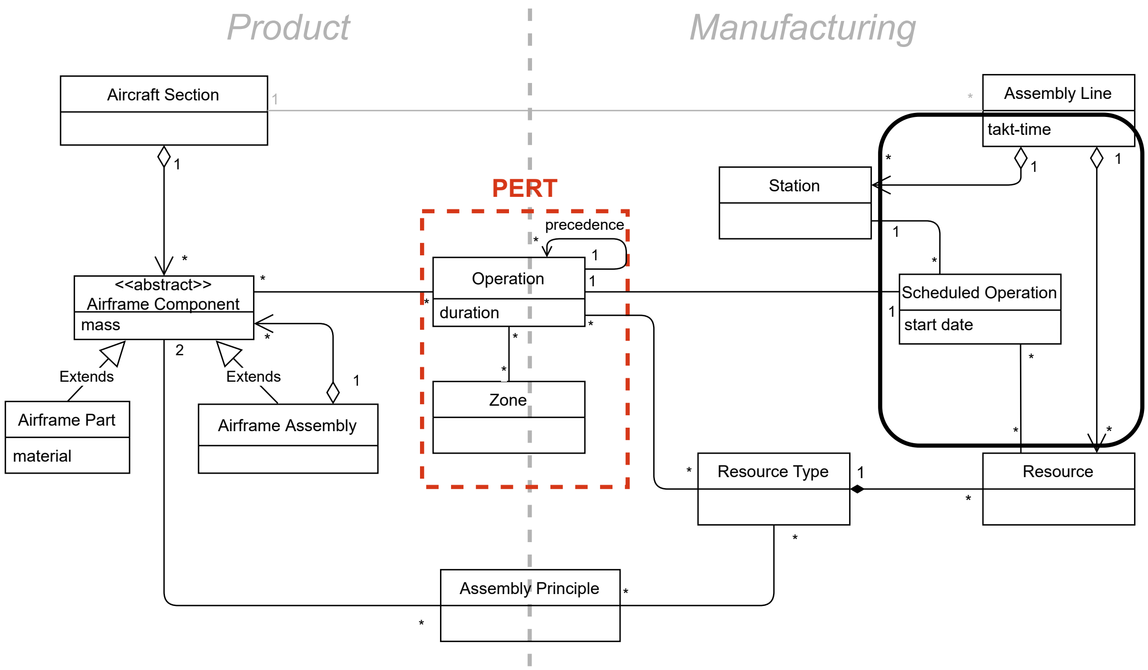

Figure 2: *SysML model for the design of an aircraft section and its assembly line*

documentation on how the metrics are computed. That way, anyone can reproduce the assessment and should obtain the same value. Finally, this assessment should not require additional work. An intuitive way to meet these four properties is to automatize the assessment.

Of course, assessing the impact of the airframe design on the assembly line is not new. Today, this activity is carried out by experts judgment. However, this expert judgment is quite arbitrary, slightly unclear and difficult to reproduce. It does not fit regarding our expectations. In addition, in our context, assessment comes to solve a highly combinatorial problem whose optimal solution is hardly findable by humans. To overcome these limitations, we propose an assessment based on automated calculus that can support experts in their design decision-making. The criteria and metrics used for this assessment correspond to the soft goals that have been described in the previous section. As these soft goals have been validated by the architects, this eases the assessment acceptation by the teams.

To formally define and to automatize the assessment, it is necessary to model the elements targeted by these criteria and also the elements that interact with them. The idea here is not to fully model the two systems, but only the parts that are relevant for expressing the problem. Modelling is playing a substantial role in engineering today, as it allows to manage complexity, formalize or run automatic reasoning on it. In our case, we have chosen a conceptual modelling expressed in SysML, as it allows the representation of elements, their properties or attributes, and their relationships.

4.2. Application: Conceptual Model

Applied to our case study, the conceptual model in Figure 2 uses the SysML formalism to describe the main elements composing an aircraft section for the product side and the main elements composing an assembly line for the manufacturing side.

On the product side, the main element is the *Aircraft Section* (the fuselage in our application). It is composed of several *Airframe Components*. An *Airframe Component* can be an *Airframe part*, *i.e.* a unitary element made in a unique material, or an *Airframe Assembly*, *i.e.* a composition of *Airframe Components* in a tree structure.

On the manufacturing side, the main element is an *Assembly Line*. An *Assembly Line* is composed of *Resources*, which represent infrastructures, machines or tooling. *Resources* are associated to a *Resource Type* that represents the type of resource used (*e.g.* drilling robots). An *Assembly Line* is also composed of *Stations*. A *Station* represents a physical space in the factory used to perform several *Scheduled Operations*. Each *Scheduled Operation* has a *start date* and uses a set of *Resources*. The duration spent on each station of an assembly line is called the *takt-time* and is the same for all the stations. So, a shorter *takt-time* means a higher rate. Note that the *takt-time*, the *lead time* and the number of stations are mathematically related by the following formula: $\textit{lead time} = \textit{number stations} * \textit{takt-time}$.

In the middle of Figure 2, there are elements associated with dependencies between the product side and the manufacturing side identified in the i^* model. The Assembly PERT is represented by two elements. The first element is the *Operation*. It has a *duration* and a link on itself to represent the precedence relation. The Assembly PERT is also linked to the *Zone*. *Zone* corresponds to a physical zone of the aircraft section where the *Operation* is performed. Each *Zone* has specific constraints with regards to working conditions that are not detailed here for readability. Each *Operation* can require *Resource Type* to be performed. It is linked to a *Scheduled Operation*. Finally, on the product side, each *Operation* is also linked to an *Airframe Component*, which is the element addressed by the *Operation*.

The *Assembly Principles* identified in Figure 1 is also present in Figure 2. Each couple of *Airframe Components* can have one or many *Assembly Principles* if they are meant to be assembled together. An *Assembly Principle* is the mean by which *Airframe Components* are connected together. For example, it can be by drilling and using bolts or by welding, *etc.* In addition, each *Assembly Principle* requires specific *Resource Types* for the assembling.

Several elements are circled in black. We call them *calculated elements*. They correspond to the tasks outputs identified in the goal model presented in Section 3. The instantiation of these calculated elements is therefore the result of an assembly architect's design task. More precisely, the task *Do the balancing of the assembly operations* corresponds to the instantiation of the class *Scheduled Operation* and to the choice of a *takt-time* for the *Assembly line*. The task *Choose the resources* corresponds to the choice of the number of *Resources* of each type in the *Assembly Line* (multiplicity value of the link between *Assembly Line* and *Resource*) and to the choice of resources for performing *Scheduled Operations*. Finally, the task *Set the number of stations* corresponds to the number of stations in the *Assembly Line* (multiplicity value of the link between *Assembly Line* and *Station*).

4.3. Application: Assembly Line Design Optimization Tool

In order to evaluate the impact of the aircraft design on the manufacturing soft goals and to support assembly line design, we have developed an Assembly Line Design Optimization tool (ALDO) that computes the best assembly line associated with this design. ALDO is based on the conceptual model defined previously: its inputs are instances of all the elements of the conceptual model, except the *calculated elements* that are its outputs.

Practically speaking, the tool explores assembly line alternatives. These alternatives correspond to the tasks *Choose the resources* and *Set the number of stations* and therefore have an impact on the soft goals. Then, the objective of ALDO is to minimize the takt-time, which contributes to

Number of stations	1 robot of each type		one additional drilling robot	
	Design 1	Design 2	Design 1	Design 2
4	9h54	**9h51**	9h54	**9h08**
5	**8h53**	9h07	8h09	**7h30**
6	**6h50**	9h07	6h47	**6h21**
7	**5h51**	9h07	5h51	**5h30**
8	**5h30**	9h07	**5h30**	**5h30**

Table 1: Takt-time *obtained for a number of stations and for some available resources.*

the goals *Have a good rate* and *Have a good lead time*. In fact, as the number of stations is fixed in the explored alternatives, minimizing the takt-time also minimizes the lead time. Based on this objective, ALDO creates a *Scheduled Operation* along with its *start date* for each *Operation* of the PERT. This corresponds to the task *Do the balancing of the assembly operations*.

The optimization problem solved by ALDO consists in scheduling operations according to several types of constraints. The first set of constraints addresses the operation precedence relation. The second deals with the fact that at most one technician can work in a zone at each time and that some zones can be blocked when performing some operations (*e.g.* for security reasons). The last set of constraints focuses on the number of available resources and the fact that some pairs of resources cannot be both installed in the same station. Concerning the objective, the tool minimizes the maximum use duration of each resource, which is an approximation of the takt-time. Following classical approaches such as [25] or [5], such a problem can be encoded in Constraint Programming. In our case, ALDO uses CP Optimizer 20.1.

We have run experiments with ALDO on benchmarks that represent two aircraft designs along with their PERT. They are composed of 150 operations on average. We consider 5 types of resources in the assembly line, mainly drilling robots and arm robots. There are approximately 20 working zones. We use two parameters to define assembly line alternatives: the number of stations and the number of resources for each resource type.

In Table 1, we present a representative set of results. The number of stations varies between 4 and 8 and we compare only two resource alternatives: in the first column there is exactly one robot of each type and in the second column there is one additional drilling robot. Note that the one robot of each type alternative helps more the soft goal *Have a minimal investment* than the one with the additional robot. The tool was given 2 minutes for each alternative, which allowed to get solutions close to optimal ones. The lead time for each alternative can be computed by multiplying the number of stations with the takt-time value.

Within the one robot of each type alternative, Design 1 has a smaller takt-time than Design 2 with a number of stations between 5 and 8 (which corresponds to a higher rate), while Design 2 is better only for 4 stations. The takt-time stops being improved after 8 stations for Design 1 and 5 stations for Design 2. It is because it is not possible to schedule in a shorter amount of time some activities that use a specific resource that must belong to a unique station. To overcome this, the assembly architect proposed a new alternative with one additional drilling robot. With this new robot, Design 2 has a strictly better takt-time for all the station numbers except for the 8 stations one in which both designs have the same value.

As a result of these experiments, we allowed architects to compare designs and to propose quickly the best possible assembly line alternatives. It should be noted that we do not obtain one best solution but a Pareto front of takt-time, lead time, number of stations and resources. Choosing among these aircraft designs and assembly line alternatives would require a trade-off among assembly line soft goals based on business knowledge. This is clearly outside the scope of this study.

5. Lessons learned

As stated in the introduction, our aim is to *take into account the performance of the manufacturing system in the design of the aircraft*. The models and tools presented in this paper clearly provide insight on the impact of certain aircraft design choices on the manufacturing system. It is important to note that the architects did not know GORE approaches at the beginning of the study. The i^* framework allowed us to elicit the key elements of assembly line design and to understand their interactions with airframe design. The resulting goal model, although simple, ensures that we did not miss any relevant elements and that their dependencies were correct. The architects considered this contribution very valuable and this opinion was shared by an expert from the company's digital transformation department to whom they showed this work. In addition, our automatic tool allows users to automatically compute optimized assembly line alternatives which would be extremely challenging to be created manually. It also shows that some alternatives appear more promising than others. The aircraft and assembly architects were very enthusiastic about the results found and the ability to measure the impact of aircraft design choices on the assembly line in a very short time. However, a limit of our approach is that it does not trace nor clearly identify the root decision in the aircraft design that has an impact on each soft goal on the manufacturing system side.

We can also observe that our interlocutors were not able to choose only one satisfactory scenario among the ones generated by the tool, but to identify some suitable alternatives. Indeed, at this stage, it was not possible for them to decide which costs should be reduced in priority, among the ones generated by the number of robots, by the number of stations, by the lead time or by the rate. However, our results allowed them to brainstorm and discuss these points in the early stages of the design, which they had never been able to do before.

Regarding the two different aircraft designs in our case study, we can note that they are equivalent in terms of aircraft architect's goals. Of course the two structure designs are not equivalent in terms of configuration and layouts, and even less identical, but they both satisfy the goals considered in the early aircraft design phase. There is therefore no trade-off between the goals associated with the aircraft design and the ones associated with the assembly line design. Practically speaking, this absence of a trade-off possibility is due to the absence of soft goals on the aircraft design side. Future work should introduce trade-offs between aircraft and manufacturing goals, such as reduce the aircraft weight and reduce the assembly time.

However, by giving the aircraft architects the possibility to compare different designs in terms of industrial performance and by allowing the assembly architect to measure the impact of choices such as the number of robots, we have laid the first steps of a methodology and tools for co-design. Historically, co-design, or concurrent engineering, aims at the early detection of potential problems between design and manufacturing [27]. Nowadays, co-design encompasses a much larger issue. It ensures that all stakeholders, not only design and manufacturing engineering, but also supply chain, maintenance, environmental impact or recycling, cooperate in the early design phases [18].

If we take a step back and look closer at the methodology we followed, we identify two steps. In a first step, we model the objectives of each system and their dependencies. These dependencies represent the point in the design activity where decisions can be made to find optimal solutions for both systems. Note that we are independent of the used GORE approach. As long as the dependencies among systems design can be represented, their rationale are detailed and their soft goals are elicited, we are confident that any GORE language can be used. In addition, even though we did not explore this direction, specification of links along requirements and interactions with them is not specific to GORE modelling. Similar modelling problems are studied in systems engineering approaches like, for instance, SysML.

In a second step, we focused on assessment and optimization. Using conceptual model, we provided a clear view of relationships among the drivers and a structure for the data. By drivers, we mean the constraints and the decision variables that affect the satisfaction of the objectives of interests. Once the problem is structured, we use automatic calculation to find optimal solutions with respect to criteria derived from the goal model.

Thus, we can observe that we couple here two different ways of modelling in order to support the assessment and optimization of solutions for our problem. Both models are complementary, bringing their own benefits at different steps of the approach.

6. Related Work

The approach presented in this paper is related to many research topics, going from the optimal design of a product and its production system to the use of multi-modelling approaches. In this section, we give a non-exhaustive view of related work.

6.1. Optimal design of a product and its production system

Many works exist in the aeronautical industry which aims at improving the manufacturing system performance. For example, at Airbus, [16] focuses on improving the existing production line relying on a collective exploratory approach. Another example at Boeing focuses on optimizing the production flow and processes, relying on modeling and simulation analysis [17]. However, few works focus on the optimal global design, even though *concurrent engineering* is a common practice in aeronautic corporations such as Airbus [1]. In recent works, Donelli *et al.* describes a value-driven model-based approach to assess a solution trade space for the aircraft design, manufacturing and supply chain [9]. Their main objective is to support decision makers in the early stages of aircraft development by coupling these different domains. Even though our objectives could appear similar, they consider a much more abstract design level. For instance, they model the dependency between the manufacturing and the aircraft with a single numeric impact factor. Another work studies the links between the factory and the aircraft [23]. They propose a conceptual model pattern to represent and apprehend the links between them but unlike us, they do not tackle the problem of finding an optimal factory. Note that the conceptual model we have used is compliant with this pattern.

There is a lot of work on *concurrent engineering* to answer our problematic, but they are not specific to the aeronautic field. For example, it is mainly applied in the automotive supply chain where it has been focusing for a long time on manufacturability issues [12]. In this approach, collaborative engineering considers mostly physical parts of a system and aims to design things that are assembled [31], or have the best assembly sequence [14, 8]. Focusing only on ease of assembly *Design for Assembly* (DFA) [4] and Design for Manufacturing (DFM) [19] aim to solve manufacturing problems at the design stage.

Stoffels and Vielhaber study a concurrent engineering approach for a product and its production system [28]. This approach is based on correlation matrices between characteristics of both systems. They also define a method in which solutions of all product functions are evaluated by experts with regards to solutions of all production functions. This human-based evaluation is done on several dimensions that can be technological, economical, ecological, *etc*.

Hanafy *et al.* [15] try to automatically compute interdependencies between product features and capabilities of machines for production. The authors use Bayesian Network on several instances of product and production systems to capture interdependencies without explicitly expressing them. This approach relies on the hypothesis that the product features and machine capabilities are already known whereas this might not be the case when reasoning at goal level. Moreover, the approach might not scale up to the complexity of an aircraft.

6.2. Trade-off between design choices in other fields

Out of the context of a product and its production system but still related to the trade-off design choices we are tackling, a method mixes goals, non-functional requirements, scoring with fuzzy rankings by stakeholders [32]. Based on this scoring, in a situation of conflicting goals, the authors introduce measurement of design solution alternativeâĂŹs influence on two goals satisfaction to help designers to make the best trade-off decision. This method focuses mainly on two goals comparisons and may not be sufficient enough to study bigger conflicts.

In another direction, Lightswitch [26] is an approach to define IT systems early requirements while taking into account the evolution of the enterprise's environment. Indeed, environment has an influence on the enterprise goals, and thus on the IT system ones, which must therefore be adapted over time. The approach stands on a process to build a model of the relationships between the enterprise and its environment, analyse and improve the way a company regulates these relationships and lastly specify the IT system goals. This approach allows to negotiate high-level goals trade-offs for the enterprise and the environment, but is very informal.

6.3. The relation with systems of systems

One could see similarities between the systems we are describing and Systems of Systems (SoSs). However we think our systems are much more dependent. Nevertheless, we can relate to some SoSs Requirements Engineering work. For instance, Ncube and Lim give perspective and research agenda [22], and they recommend some research topics very close to our work. The first topic is"*Tools for SoSs Requirements Trade-Off Decision*" where the authors advocate the development of techniques and tools to permit efficient trade-off decisions among a large trade space. The second topic is "*Multi-Level Modelling techniques for SoSs requirements*", where the authors discuss different modelling approaches, goal modelling included, and prone combination of approaches to take advantage of different perspectives.

6.4. The use of multi-modelling approaches

Some works advocate the use of multi-modelling techniques for Requirements Engineering. For instance, Franch *et al.* study the joint use of i^* with other modelling techniques [10]. In this context, they reviewed existing work and identified different scenarios of use. Our work belongs to a scenario of Model Coupling, where the goal model and other modelling notations coexist without being merged, in order to gain benefits. Closer to our approach, Alencar *et al.* go from an i^* model to a conceptual model in UML [2], but we differ on the final objectives. Indeed, we are dealing with the development of design-dependent systems and their optimization while they are interested in going from a goal model to a UML representation for software development. Their work is mainly centred on the model transformation guidelines.

7. Conclusion and Future Work

In this paper, we have presented methods and tools to support the design of an aircraft and its manufacturing system in order to take into account the impacts of the former on the latter. We have provided an approach that combines goal modelling, conceptual modelling and constraint programming in order to automatically compute optimized solutions applied to the design of an aircraft fuselage airframe and its assembly line. This approach has been very well received by the architects and has allowed them to discuss design alternatives at an early stage in the design.

Among the future work, we have already identified some of them in the lessons learned. We have observed that there is no possible trade-off between the goals impacting the aircraft designs and the ones impacting the assembly line, due to the absence of soft goals on the aircraft design side in our case study. One future work could be to introduce them to the problem in order to enable and explore this kind of trade-off. New case studies will be an opportunity to test our approach and clarify how it should be used.

The improvement of the optimization tool is also a possible future work. We could explore the generation of explanations for the values obtained on the criteria and link them to particular aircraft design decisions, in order to guide the architects in a better way.

We could also strengthen the approach by investigating in detail the relationship between the two modelling techniques we have used. In this direction, we could try to establish guidelines to express how to progress from one model type to the other. In addition, in a model-based approach perspective, we could explore the possibility to return the computed solutions in the form of a complete conceptual model instantiation.

Finally, we could generalize our work to other systems which are designed in an asynchronous way and are dependent on each other. As we have seen in Section 1, this kind of systems can be found in other domains and may benefit from our approach. In this context, we have just started working on a case study coming from the space industry.

References

[1] Airbus. Design webpage. Last accessed 17 March 2022.

[2] Fernanda Alencar, Beatriz Marín, Giovanni Giachetti, Oscar Pastor, Jaelson Castro, and João Henrique Pimentel. From *i** requirements models to conceptual models of a model driven development process. In *Proceedings of PoEM 2009*, pages 99–114. Springer, 2009.

[3] Daniel Amyot, Sepideh Ghanavati, Jennifer Horkoff, Gunter Mussbacher, Liam Peyton, and Eric S. K. Yu. Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 25(8):841–877, 2010.

[4] Geoffrey Boothroyd. Product design for manufacture and assembly. *Computer-Aided Design*, 26(7):505–520, 1994.

[5] Tamara Borreguero Sanchidrián. *Scheduling with limited resources along the aeronautical supply chain: from parts manufacturing plants to final assembly lines*. PhD thesis, E.T.S.I. Industriales (UPM), 2019.

[6] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*, volume 5 of *Int. Series in Software Engineering*. Springer, 2000.

[7] Fabiano Dalpiaz, Xavier Franch, and Jennifer Horkoff. istar 2.0 language guide. *CoRR*, abs/1605.07767, 2016.

[8] F. Demoly, Xiu Yan, B. Eynard, L. Rivest, and S. Gomes. An assembly oriented design framework for product structure engineering and assembly sequence planning. *Robotics and Computer Integrated Manufacturing*, 27(1):33–46, 2011.

[9] Giuseppa Donelli, Pier Davide Ciampa, BjÃűrn Nagel, GlÃľverson Lemos, JoÃčo Mello, Ana Paula C. Cuco, and Ton van der Laan. A model-based approach to trade-space evaluation coupling design-manufacturingâĂŞsupply chain in the early stages of aircraft development. In *AIAA Aviation Forum*, volume 2021-3057, 2021.

[10] Xavier Franch, Alejandro MatÃľ, Juan C. Trujillo, and Carlos Cares. On the joint use of i^* with other modelling frameworks: A vision paper. In *Proceedings of RE 2011*, pages 133–142, 2011.

[11] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.

[12] Ingrid Göpfert and Matthias Schulz. Logistics integrated product development in the german automotive industry: current state, trends and challenges. In *Proceedings of LDIC*, pages 509–519. Springer, 2013.

[13] Renata S. S. Guizzardi, Feng-Lin Li, Alexander Borgida, Giancarlo Guizzardi, Jennifer Horkoff, and John Mylopoulos. An ontological interpretation of non-functional requirements. In *Proceedings of FOIS*, volume 267 of *Frontiers in Artificial Intelligence and Applications*, pages 344–357. IOS Press, 2014.

[14] Riadh Ben Hadj, Imen Belhadj, Moez Trigui, and Nizar Aifaoui. Assembly sequences plan generation using features simplification. *Advances in Engineering Software*, 119:1–11, 2018.

[15] Mohmmad Hanafy and Hoda ElMaraghy. Co-design of Products and Systems Using a Bayesian Network. *Procedia CIRP*, 17:284–289, 2014.

[16] Honorine Harlé, Sophie Hooge, Pascal Le Masson, Kevin Levillain, Benoit Weil, Guillaume Bulin, and Thierry Ménard. Innovative design on the shop floor of the Saint-Nazaire Airbus factory. *Research in Engineering Design*, 33(1):69–86, 2022.

[17] R.F. Lu and S. Sundaram. Manufacturing process modeling of Boeing 747 moving line concepts. In *Proceedings of the Winter Simulation Conference*, volume 1, pages 1041–1045, 2002.

[18] Y-S Ma, Gang Chen, and Georg Thimm. Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. *Journal of Intelligent Manufacturing*, 19(6):625–641, 2008.

[19] E. Molloy, H. Yang, J. Browne, and B.J. Davies. Design for assembly within concurrent engineering. *CIRP Annals*, 40(1):107–110, 1991.

[20] John Mylopoulos, Lawrence Chung, and Brian A. Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483–497, 1992.

[21] John Mylopoulos, Lawrence Chung, and Eric S. K. Yu. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1):31–37, 1999.

[22] Cornelius Ncube and Soo Ling Lim. On systems of systems engineering: A requirements engineering perspective and research agenda. In *Proceedings of RE 2018*, pages 112–123. IEEE Computer Society, 2018.

[23] Thomas Polacsek, Stéphanie Roussel, François Bouissière, Claude Cuiller, Pierre-Eric Dereux, and Stéphane Kersuzan. Towards thinking manufacturing and design together: An aeronautical case study. In *Proceedings of ER 2017*, volume 10650 of *Lecture Notes in Computer Science*, pages 340–353. Springer, 2017.

[24] Thomas Polacsek, Stéphanie Roussel, Cédric Pralet, and Claude Cuiller. Design for efficient production, A model-based approach. In *Proceedings of RCIS 2019*, pages 1–6. IEEE, 2019.

[25] Cédric Pralet, Stéphanie Roussel, Thomas Polacsek, François Bouissière, Claude Cuiller, Pierre-Eric Dereux, Stéphane Kersuzan, and Marc Lelay. A scheduling tool for bridging the gap between aircraft design and aircraft manufacturing. In *Proceedings of ICAPS 2018*, pages 347–355. AAAI Press, 2018.

[26] Gil Regev and Alain Wegmann. Defining early IT system requirements with regulation principles: The lightswitch approach. In *Proceedings of RE 2004*, pages 144–153. IEEE Computer Society, 2004.

[27] Delavar G Shenas and Sepehr Derakhshan. Organizational approaches to the implementation of simultaneous engineering. *International Journal of Operations & Production Management*, 14(10):30–43, 1994.

[28] Pascal Stoffels and Michael Vielhaber. Methodical support for concurrent engineering across product and production (system) development. In *Proceedings of ICED 2015*, volume 4, pages 155–162, 2015.

[29] Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proceedings of RE 2001*, pages 249–262. IEEE Computer Society, 2001.

[30] Eric SK Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE 97*, pages 226–235. IEEE, 1997.

[31] Xuan Fang Zha, He Jun Du, and Jin Hao Qiu. Knowledge-based approach and system for assembly oriented design, part i: the approach. *Engineering Applications of Artificial Intelligence*, 14(1):61–75, 2001.

[32] Xuan Zhang and Xu Wang. Tradeoff analysis for conflicting software non-functional requirements. *IEEE Access*, 7:156463–156475, 2019.