HAL
open science

# The need of diagrams based on Toulmin schema application: an aeronautical case study

Thomas Polacsek, Sanjiv Sharma, Claude Cuiller, Vincent Tuloup

## To cite this version:

Thomas Polacsek, Sanjiv Sharma, Claude Cuiller, Vincent Tuloup. The need of diagrams based on Toulmin schema application: an aeronautical case study. 2020. hal-02468175

**HAL Id: hal-02468175**

**https://hal.science/hal-02468175v1**

Preprint submitted on 5 Feb 2020

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

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

# The need of diagrams based on Toulmin schema application: an aeronautical case study*

Thomas Polacsek & Sanjiv Sharma & Claude Cuiller & Vincent Tuloup†

**Abstract**

*In this article, Justification Diagrams are introduced for structuring evidence to support conclusions that are reached from results of simulation studies. An industrial application is used to illustrate the use of the Justification Diagrams. Adapted from the Toulmin schema, the aim of Justification Diagram is to define a comprehensive, auditable and shareable notation to explain the results, the input data, the assumptions made and the techniques applied, to construct a cogent conclusion. Further, the Justification Diagrams provide a visual representation of the argument that aims to corroborate the specified claims, or conclusions.*

*A large part of this work is based on the application of the Justification Diagrams in the context of the European project, TOICA. The Justification Diagrams were used to structure all justifications that would be needed to convince an authority that a simulation process, and the associated results, upheld a particular conclusion. These diagrams are built concurrently in a product development process that accompanies the various stages of Verification and Validation (V&V) and where, for each design stage of V&V, argumentation is constructed by aggregating evidence and documents produced at this design stage.*

## 1. Introduction

In many applications (like aerospace, embedded systems, medical devices or industrial simulations) gaining confidence in a product is achieved by performing Verification and Validation (V&V) to establish its origins and fitness for purpose. Furthermore, in the context of engineering simulation, a new notion has emerged: the concept of accreditation. According to the United States Department of Defense [2], Verification and Validation are processes for gathering and evaluating evidence, whilst Accreditation consisting of an "*official certification that a model, simulation, or federation of models and simulations and its associated data is acceptable for use for a specific purpose*" [50]. The key point is to gather and to evaluate the evidence in the context of this specific purpose. Then Accreditation activities augment the V&V activities and involves an [independent] authority who certifies the appropriateness of a model, or simulation, for the given context. For this, it is necessary to have comprehensive information (*the evidence*) that explains, for example, the results, the input data, the assumptions, the techniques used, and the conclusions reached. Therefore, the accreditation task comprises, firstly, to construct an acceptable argument for the given purpose, and secondly to evaluate the evidence for supporting, or collapsing, the argument. If the evidence supports the argument then the models, or simulations, are certified as being fit for attaining the stated purpose.

Of course, to perform certification activities we do not need to be an expert in all areas covered by the V&V processes; however, we do need consensus from a group of experts that the evidence

*The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°604981.

†Authors version, The need of diagrams based on Toulmin schema application: an aeronautical case study, pp 257-282, EURO Journal on Decision Processes, November 2018, Volume 6, Issue 3-4, Springer 2018, Print ISSN 2193-9438, Online ISSN 2193-9446

is necessary and sufficient for the stated purpose. As a simile, consider an engineer who wants to know whether or not Fermat's conjecture is a theorem. We know that Andrew Wiles propounded a proof of the theorem; it is likely that an engineer may not be an expert in number theory to understand the proof. However, (s)he can use Wiles's results with confidence because sufficient experts have checked the proof and it had been tested through a system of peer review. With this evidence, it seems reasonable to consider that Wiles's proof is correct and hence that Fermat's conjecture is indeed a theorem. Without being an expert in number theory, the engineer can be confident in reaching this conclusion.

Likewise, to be confident in the result of a design study, an assessment or a decision, we do not need to be an expert in all the fields; however, we do need to be confident that the right people have been involved in the production and structuring of the evidence. Therefore, we need to have comprehensive information comprising the results, the input data, the assumptions made and the techniques applied. Additionally, we also need the meta-data; for example: who did the work, what was done, why it was done, when, how and where it was done. Hence we need to identify the list of key evidence, to construct, assess and evaluate the justification to reach cogent conclusions. As such, in this work we focus on structuring the evidence to construct a convincing justification rather than on the accreditation aspects.

To account for that, in [36] Polacsek has introduced the notion of "*Justification Diagrams*" to structure all justifications needed to convince an authority that a product provides pertinent answers to particular questions. The aim of this diagram was to communicate a convincing and a valid argument for specified claims. We note that even we use the term argument, our proposal is not in the same vein as the classical works in argumentation theory [3]. Indeed, in general most of the works, following this theory, are focused on how to resolve conflicts between a set of predefined or constructed arguments either by relaying on the seminal work of Dung [10] or by adopting a dialectical process through argumentation- based dialogues [39]. In this work, we propose to use Justification Diagrams in an industrial case study. More precisely, we extended the use of this diagram to structure a clear understanding of the evidence, produced by the modelling and simulation processes, to reach justifications that a multi-disciplinary team would deem cogent.

We adapted the Toulmin schema [49] to construct the Justification Diagram. The Justification Diagram structures the V&V information to substantiate why a decision-maker can be confident in using the results from modelling and simulation processes. The Justification Diagram utilises diagrammatic representations to structure the formal and informal information that contribute to justifying the results. It is still the aim that these representations provide a comprehensive, auditable and shareable notation for displaying this information for a multi-disciplinary team. Note that a set of these diagrams are intended to be constructed concurrently with a product development process, such that for each stage of V&V, justifications are constructed by progressively aggregating evidence and information produced at that stage.

The need for argumentation to explain complex results is not new. The rise of intelligent systems in the early nineties led to many publications that emphasised the need of explanations to enable humans to accept automatically generated results [1, 5, 23, 46] (note that the question is still topical [32]). These published cases reported that explanations of results, based on logical reasoning, created a positive perception in the users and increased their confidence in using these systems. Motivated by these works, we present explanations of results in diagrammatic form. We extend the explanation of computational results to explain results arising from complex process, specifically where humans need to interact with these results.

The layout of this article is as follows: after a brief state-of-the-art on argumentation in section 2, we then introduce our generic argumentation pattern in section 3. Section 4 focuses on the implementation of the Justification Diagrams on an industrial study case and section 5

contains our conclusions.

## 2. State of the Art

### 2.1. Examples of models for Diagramming justifications

Almost all human cognitive activities use explicit conceptual models. In engineering, they are useful to not only define, explain and reason about a system, but also to support the information creation and decision-making processes. In many projects today, model-based engineering is an essential approach for navigating through the complexities of multidisciplinary product designs. Complexity also arises due to the different types of models used to represent the various aspects of a product and perspectives of the multitude of stakeholders. These models have specialised notations suited to the separate viewpoints of the stakeholders. For example, in System Engineering there are models for business process (e.g. BPMN [33]), for system behaviour (e.g. State machine [21]), for system structure (e.g. class diagram [44]) and for requirements (e.g. KAOS [51]). These models need to be consistent and coherent with each other to enable human-centric activities like the organisation of knowledge (as described in [16]) and decision making (like IBIS [26]).

Exemplars of models for organising knowledge include Issue-Based Information Systems (IBIS). IBIS addressed underpinning and documenting the production of knowledge. It defined three key elements: *Questions*, *Ideas* and *Arguments*. The Questions are elements representing all questions related to the project; the Ideas are answers to the Questions of fact and the Arguments are opinions (positions and beliefs). In the case of IBIS, numerous notations have emerged to address the design process creation [29]. The two common aspects of models for organising knowledge are a) the capture of justifications leading to the decisions taken during the design, and b) the different design alternatives to be taken into account in the decision-making. However, all these IBIS like approaches are used for selecting a design amongst a number of design choices during the early stages of development.

In this work, we address a different, but related problem. We are interested in constructing the justification for accepting a product; therefore, we want to keep track of the alternatives and to explain the reasons for the acceptability of a set of products, at the appropriate level of granularity. Rather than addressing the consistency of a product directly, we use the Justification Diagram as a means of upholding the justification if the properties of a product are consistent.

Indeed, the Justification Diagram shows clearly the inputs, the outputs and evidence in an auditable structure. The Justification Diagram contributes to the understanding of the rationale of V&V activities; equally well for the authorities, for the review committees and for other stakeholders. We claim that having such a diagram allows navigation among the information and eases the identification of areas where there is a lack of evidence. Further, if evidence were to change, then it is easier to assess the impact of the change on the overall case for justification. Finally, the use of the Justification Diagram also contributes to the ideas of modularity and of reusability.

### 2.2. Toulmin schema

According to Reed et al.[41], the use of diagrams that support reasoning is well established with a long history. For example, Richard Wately in his book "*Elements of Logic*" (1836, pp. 420-430) describes diagrams used for illustrating an argumentation. Wately provides a diagram (page 342) with the following narrative: "*Many students probably will find it a very clear and convenient*

*mode of exhibiting the logical analysis of a course of argument, to draw it out in the form of a Tree*". In another words, the diagram is not useful for the logical demonstrations but helps the students to understand them. In 1917, Wigmore defined a visual representation for structuring hypothesis leading to evidence in the legal framework as presented in [14]. For more information, see [41] for a discussion on the history of argument diagramming.

In 1958, Toulmin defined a model with six characteristics that make a good, convincing argumentation (see [49] for details). Many American universities teach this model to explain the mechanisms of the argument [56, 22], for example, in legal argumentation theory and critical thinking modules [18]. Today, the acceptability of an argument and the underlying mechanisms is studied by a variety of disciplines including computer science (through artificial intelligence) [53, 3], linguistics [30], epistemology [31] and the legal sciences [35].

The Toulmin schema focuses on ways of drawing conclusions from evidence. Toulmin used supplementary data that he called *Warrant*, to justify the reasoning from set of evidence to a conclusion. The Warrant is the attribute that legitimises the passage from the evidence to the conclusion; it clearly explains the inference of a conclusion. Moreover, Toulmin adds to his schema *Qualifier* in order to express reservations because a conclusion is not always true. The Qualifier corresponds to notions of possibilities or probabilities. The Toulmin schema also contains *Rebutal*, the circumstances under which the conclusion is not true, and *Backing*, which binds justifications or credential to the Warrant.

Therefore, starting from ToulminâĂŹs pattern we propose a design of a generic Justification Diagram orientated towards industry applications. Note that in the Toulmin schema, the set of Supports is a set without relation amongst its elements. There is no hierarchy amongst the Supports, nor any dependency. In reality, things are more complex. It is possible to identify ancillary evidence where one, or more, pieces of evidence reinforces another, as in the case when an expert guarantees a property and another expert guarantees that the first expert is qualified to provide the guarantees. Unlike Toulmin, many works [54, 26, 55, 40] studied distinctions among the Supports and their relations with a claim, but it seems difficult in our case to make such king of distinction. Indeed, our aim is to propose a simple and intuitive model for engineers and experts to support them in organising and to auditing a mass of justifications. If the fine characterisation of the Supports were important for understanding the argumentative reasoning, then, unfortunately, it would carry a significant overload of work in our context.

## 2.3. Diagramming and safety cases

We could link the Justification Diagram with the concept of safety cases. In the context of risk assessment and dependability, a safety case is a structured set of information that provides a set of justifications on whether a system fulfils its safety requirements. The Safety Management Requirements for Defence Systems states that [8]: "*a Safety Case shall consist of a structured argument, supported by a body of evidence*". Furthermore, ISO/IEC 15026 [24] recommends, without explaining how, to structure a set of assertions and statements (goals and sub-goals) to have an argumentation for high-level claims. Even if there were no standards for safety case diagrammatic representations, there is a trend to provide a safety case as a graphical representation by using the *Goal Structuring Notation* (GSN). GSN is a notation based on the work of Kelly and Weaver [25]. The notation is designed to be an effective means for presenting, maintaining and reusing safety arguments. It is mainly a graphical notation dedicated to dependability. GSN is oriented towards decomposition; the user starts from the conclusion, with the aim to justify, and then to breakdown the goal into more simple assumptions to obtain, at the end, evidence.

However, GSN is fully safety oriented and was not designed for generic use. The aim of a

GSN diagram is to demonstrate, in a graphical form, that safety properties are satisfied and risks have been mitigated. We want to design a Justification Diagram, not just for safety (although there is extensive literature on the use of GSN for wider applications); however, GSN is too safety oriented for our case. Indeed, even if historically the notation is based on Toulmin schema, GSN is no longer considered to be directly related it [4]. In GSN, strategies are decompositions and the approach is clearly top-down. More importantly, Toulmin schema treats Guarantee as the cornerstone of the reasoning, whilst it is only an optional element in GSN. Finally, GSN is more a notation than a formal study of the argument.

## 2.4. Diagramming tools

Focusing on software tools to implement the Justification Diagrams, at the beginning of this work, we surveyed existing software such as Carneades [15], Argumed [52], Araucaria [42] and IBIS. Carneades is a software implementation based on a formal argumentation model; however, it provides evaluation capability based on pre-established schemas. In the field of legal argument, Verheij [52] defines a Non-Monotonic Logic[1] to model an argument and proposed Argumed as an implementation tool. Also, in legal field, Araucaria [43] performs text analysis and assists the user in the construction of an argumentation scheme as a tree structure, the software was tested by judges in Ontario, Canada.

Based on the same idea, there are many graphical tools to visualise and represent argument, including Argunet, Compedium and bCisive. However, none of these tools has been able to satisfy our application needs and expectations. Indeed, on one hand, tools based on academic works are very useful and help for understanding argumentation theory, but they are too complex and use too many concepts to be used by engineers in their daily work. Learning to use these tools can be too laborious and many concepts are too specialised for our use. On other hand, simply graphic tools do not have sufficient semantics to be unambiguous; they do not lend themselves for a systematic capture of rational.

Based on these observations we returned to the work of Stephen Toulmin and constructed a set of diagrammatic representations, which are described in the next section. Note that the semantic definition of these representations is an ongoing work.

## 2.5. The need for explanation

Providing an explanation to convince that a result is well found does not arise only in certification context. Indeed, the use of intelligent systems, such as Expert Systems, Decision Support Systems, forecasting systems and decision support tools, is also hampered by the need to convince that the results given by the automatic system are correct [47].

For example, we can cite the work of Chandrasekaran [5] and Southwick [46] who propose to cluster the explanations provided by such systems into three categories:

***Trace Explanation*** provide an auditable chain consisting of the information used, the rules applied and the steps taken by the system to produce the result. These Trace Explanations, answer the question "how were the results produced?"

***Strategic Explanation*** explain why one piece of information is used before another; how the different stages of reasoning are selected and how they contribute to achieve the main goals. Strategic Explanations provide an explanation of the overall functioning of the system.

[1]Non-Monotonic Logic is a formal frameworks devised to capture and represent kinds of inference in which reasoners draw conclusions tentatively, and can reserve the right to retract them in the light of further information.

***Deep Explanation*** provide justification to the foundations of intelligent systems. These justifications provide the theories from which the result was generated, the underlying logics, and the rationale of the knowledge base.

Using an empirical study, Ye [57] showed that providing explanations, more precisely deep explanations, is a prerequisite to accepting the audit advice produced by an expert system. In the same vein, using a different empirical study, Lawrence [27] showed that providing an explanation is the most effective method to improve the acceptance of results from a forecast system; in this case they did not make any distinction on the types of explanations described above. Ye [58], found that Deep Explanations were the ones that led to the best acceptance of the system results by users. Further, Gönül [12] found that the acceptability also depended on the expertise of the users involved.

In addition to the type of the explanation, the meta-study performed by Gregor [19] showed that the design of the explanation is also an important factor. Through the survey of these studies, they reported that the provision of justifications, preferably a though Deep Explanation, is necessary; further, in order to be particularly effective, the provision of justifications must conform to the Toulmin scheme. Note that, [57] had already made a direct link between explanation and the work of Toulmin; as evident through the three types of explanations in the form of a Toulmin diagram.

Justification diagrams are not intended to answer the problem of explanation in intelligent systems. However, we could imagine in the future that such diagrams could be used as a support to present intelligent system explanations to users. A software program could thus clearly present the rationale behind its result. Similarly, justification diagrams could be a response element for "*right to explanation*" [13].

## 3. Our model: a Toulmin schema variation

The two main industrial needs that we aim to address with the Justification Diagram are:

1. to structure evidence for justifying conclusions that arise from results of modelling and simulation processes; and
2. to provide visual representations for stakeholders from disparate disciplines to have a common platform for reasoning about the justification.

As such, to address the first need, we define a model comprises justification and their relationships. For the second need, we proposed a diagrammatic representation of our generic justification pattern.

### 3.1. The justification diagram

For our justification model, we drew inspiration from the ToulminâĂŹs argumentation model [49], with focus on the spirit of it, as opposed to its strict adherence. Our model comprises information objects, and their relationships, that are needed for structuring the justifications. These information objects are the result of purposeful experimentations conducted in various industrial case studies [36, 9]; one such case study is described in section 4.

In our schema, see Fig. 1, like Toulmin we have a *Conclusion* based on facts, on Supports. Looking more closely at the support of a conclusion, pre-established truths are the bases of any demonstration or proof – these truths are the accepted propositions. For example, a mathematical

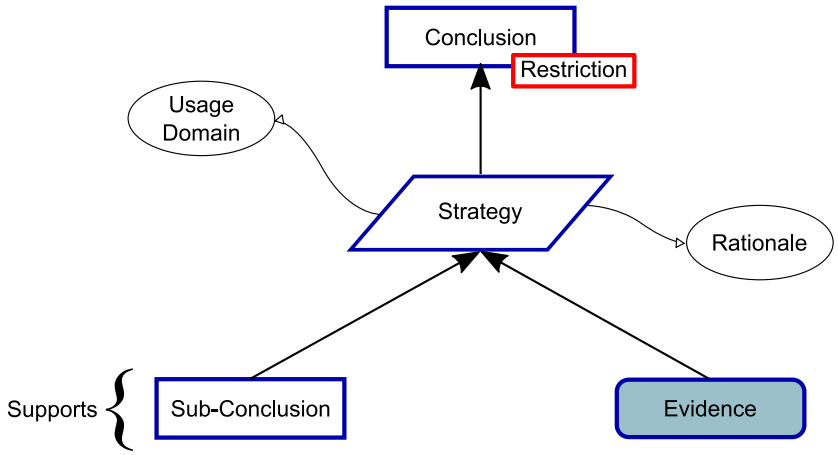


**Figure 1:** *Generic justification pattern*

proof always assumes a set of axioms to be true. These axioms require no further justifications; they are statements or propositions that are regarded as established, accepted, or self-evidently true. Similarly, the basis of any argument is a set of propositions that are accepted as true by the one who sets out the demonstration; their audience accept the truism.

In our case, the supports for a Conclusion fall into two categories. The first is *Evidence*, e.g. results published in a scientific paper, a piece of information provided by an expert, a set of practices defined by standards, or results of calculations (like numeric simulation or model checking). The second is *Sub-conclusion*, e.g. a conclusion derived from another justification step. Note that, the Sub-conclusions may have similar patterns supporting it; the process would end with only the Evidence items as the supporting information.

As in the Toulmin schema, the *Strategy* (the Warrant in Toulmin schema) is the cornerstone of reasoning in our model; it is the rationale for legitimising the transition of the Evidence to the Conclusion. For Toulmin, the Strategy clearly explains how a conclusion is inferred from data. We find the word Strategy in ISO 15026 [24] and in GSN. For ISO 15026, it is the glue that cements all assumptions together to support a conclusion and, for GSN it explains the inference of the goal from the supporting elements.

In the context of accreditation or certification, a conclusion is acceptable or not. A conclusion is true all the time, and not "*usually*" or "*sometimes*", so, we remove the ToulminâĂŹs modal qualifier. We adopt the notion of Backing by renaming it *Rationale*. As for Toulmin, we consider that in addition to the Strategy, a reasoning must provide the reasons why a specific Strategy is applicable. We add a new concept the *Usage Domain*, which provides the context for the application of a given Strategy. Finally, we change the concept of Rebuttal into a Restriction that simply expresses a limitation of the conclusion.

## 3.2. Representation of the justification diagram

We proposed a diagrammatic representation of our generic justification pattern built using the notations and elements described in Table. 1. The representation of these elements and their inter-relationships is loosely inspired by the GSN. The GSN defines two types of relationships; *SupportedBy* and *InContextOf*; both of these relationships are dyadic links. The *SupportedBy*

**Table 1:** *Elements of the generic justification pattern*

| Representation | Definition |
|---|---|
| Conclusion | *Conclusion* to demonstrate. |
| Evidence | *Evidence* is Fact, data, hypotheses, etc. The conclusion is based on it. |
| Sub-conclusion | *Sub-conclusion* is an evidence of the conclusion, but also a conclusion of another justification step (conclusion of another application of the generic justification pattern) |
| Strategy | *Strategy* corresponds to the method used to establish the connection between the supports (evidence and sub-conclusion) and the conclusion |
| Software Strategy | *Software Strategy* is a very specific strategy, where the conclusion is only the result of a computation and no human is involved. |
| Restriction | *Restrictions* are limitations of the conclusion. They are separated from the conclusion because they have no vocation to appear in the final/high level conclusion. When we explicitly add a restriction to the conclusion, it is because we think that it will be lifted in the future. An example might be a product restriction that should disappear in the future. |
| «rationale» Rationale | *Rationale* is an explanation of why a method (a strategy) is applicable here. Rationale is bound to the Strategy. It supports the method with arguments; it details the justifications for why a method is acceptable. If a strategy is to follow a process defined in a standard, then the strategy is the protocol and the rationale is the standard itself. |
| «usage domain» Usage Domain | *Usage domain* gives the precise conditions of use and the limitations of the strategy. If we take the example of the application of a standard then the usage domain will describe in which context and for which purposes this standard is applicable. |

relationship had Goal as one of its nodes and the link declares that there is an inference between a Goal and its argument, or that an Evidence substantiates a Goal. The *InContextOf* relationship provides a context under which an element applies. The GSN Standard [20] contains a fuller description of the GSN method.

For the legibility aspects of the diagrams, we decided to split the Strategy into two cases: *Software Strategy* and *Strategy*. Software Strategy refers to the use of particular software to compute a conclusion (for example, the result of a simulation or a property proved by model checking) and Strategy refers to all other cases (for example a committee, an expert analysis or the application of a normative process).

The *Goals* in GSN can be thought of as *Conclusions* in our Justification Diagram. For ease of readability, we selected the *SupportedBy* relationship because each element in our model aims to support the conclusions (see Fig. 1). Also for readability, the *InContextOf* relationship is reserved for the *Rationale* and *Usage Domain* elements. Arguably, *Restrictions* may be considered contextual; however, for this case we informally attach it to the conclusion; with the understanding that over the course of a product development process, additional work will have been performed that would remove the restriction.

In this section, we described the Justification Diagram; a diagram that captures the rational, logical and structures of all evidence that leads to accepting a high-level property. In a Justification Diagram, the conclusion is on the top of the diagram, the leaves are the evidence (data, fact, etc.) positioned on the bottom. A Justification Diagram is a structured way for representing the set of elements that lead to a conclusion. In the next section we illustrate the use of the Justification Diagram using an industrial case study.

## 4. Application

*Thermal Overall Integrated Conception of Aircraft*[2] (TOICA) was a European project with 32 partners from eight countries. TOICA completed in September 2016. The purpose of TOICA was to implement and demonstrate flexible, integrated multi-level, multi-disciplinary modelling and simulation approaches for performing design studies of aircraft architectural concept. Further, the TOCIA partners developed new capabilities to address collaborative analyses for converging on more thermally optimised aircraft concepts. TOICA had the following four high-level objectives:

1) Develop customised collaborative and simulation capabilities improving the generation, management, and maturity of the evolving Behavioural Digital Aircraft dataset

2) Develop new concepts for improved thermal load management for aircraft components, systems or equipment, which will integrate innovative cooling technologies and products

3) Assess and validate the developed capabilities and technology concepts against different common reference aircraft, targeting both Entry Into Service (EIS) 2020 and EIS 2030+ Thermal Concept Aircraft

4) Optimise aircraft design by enabling highly dynamic allocation and association between requirements, functions and product elements for creating novel products

The full description of all these aspects of TOICA is beyond the scope of work. We focus on the application of the Justification Diagram and simply state that the independent monitoring

[2] `http://www.toica-fp7.eu/`

officer agreed that the development of the Justification Diagrams contributed to objectives 1) and 2); their application contributed to objectives 3) and 4).

In the following sub-sections, we present the background to the case study to illustrate the utility of the Justification Diagram. Before describing the case study and the associated Justification Diagram, we define the enabling concepts: Architects' Cockpit, Multi-disciplinary decision making and Super-Integration.

## 4.1. Preface to the application

The design of a complex product such as an aircraft requires contributions from many different disciplines, which may often be distributed in the extended enterprise. Managing the interactions amongst results that arise from these different disciplines is necessary to ensure that decisions are based on evidence. At a human level, these interactions include negotiations amongst the engineering disciplines (e.g. aerodynamics, loads, safety and thermal), business disciplines (e.g. finance and procurement) and industrial disciplines (e.g. manufacturing and logistics). It is the role of the *architects* and the *chief engineers* to manage these interactions to drive convergence towards the targeted performances of the aircraft, the manufacturing performances and business needs [38]. In fact, the goal of this decision-making team is to ensure that the aircraft will address the various stakeholders needs; for example, to meet the operational performances while being produced at the required rate and to the targeted cost.

Fundamentally, the design of the aircraft and its production system is the result of iterative design activities performed by the various disciplines across the extended enterprise (e.g. formed through partnerships with component manufactures and equipment providers). The sequencing of these activities is organised at an enterprise-level by a Product Development Process, or more generally, a set of Systems Life-cycle Processes (see, for example, [48] for a fuller description). In order to synchronise the results of the multi-disciplinary design activities, these processes are punctuated by a series of strategically scheduled design reviews. The nature of these reviews it to assess the maturity of the design and to assure its readiness to proceed to the next stage; therefore, these design reviews will be referred to *Maturity Review*.

Further, the multi-disciplinary design activities provide answers to enable critical assessments of the design's maturity. These answers are used by the decision-making team to converge on optimal design selection; seeking optimality with respect to the product performance perspective, but also, for example, from the financial and business perspectives. Therefore, the answers provide information for the decision-makers to trade-off competing design characteristics and to specify the desired properties, capabilities and behaviours of the product. As such, these multi-disciplinary design activities will be referred to as *Trade Studies*.

Consider an illustrative example where a trade study from two disciplines generated competing characteristics: suppose that an aerodynamic part of the study required high precision surface finish to achieve some aerodynamic performance capability. Similarly, the manufacturing part of the study determined the cost of manufacture for a given surface finish. However, from a business viewpoint, it may not be desirable to have high precision manufacture and maintain an acceptable cost of manufacture. Therefore, a trade-off decision would be made to find an appropriate balance of the manufacturing precision (hence a lower cost of manufacture) whilst maintaining an acceptable aerodynamic performance.

## 4.2. Architects' cockpit

The management of these trade studies requires an entire and consistent dataset, although not necessarily complete, that describes the multiple aspects of the interactions amongst the

main aircraft components. The user-facing system that provided this consistent and contextual information to allow the architects to interact with this information was called the *Architects' Cockpit*. The main aim of the Architects' Cockpit is to be a central collaborative environment that provides aircraft architects the ability to:

1. define and manage architectural definitions, which includes
   (a) select candidate aircraft configurations for trade studies
   (b) visualise the different aircraft configurations
   (c) create and associate requirements, e.g. as defined by the desired operations of the aircraft
   (d) display of the functional or physical views and their associated relationships
2. set-up and orchestrate trade studies, as described in the preceding paragraph, including
   (a) identify target performance characteristics and key parameters for evaluation
   (b) define the trade study and the set of analyses that would produce the results for evaluation
   (c) allocate steps of the trade study to the various disciplines and organisations (where the extended enterprise is involved)
   (d) launch the trade study requests and monitor their progress towards completion
3. collect and integrate results from the various engineering, business and industrialisation disciplines, thereby
   (a) analyse and assess the results in a collaborative, multi-disciplinary maturity review event
   (b) perform trade-off assessments
4. evaluate the results, take decisions and record them
   (a) evaluate candidate solutions and decide on the best in class
   (b) record the justification for the rejected solution as as well as the accepted solutions
   (c) decide on the next course of action to evolve the design further.

Essentially, the Architects' Cockpit is the architects' "window" into the evolving aircraft dataset; from the initiation of the aircraft definition to the decisions at the maturity review events, throughout the life-cycle of the design.

Regarding item 4 in the list, the analyses of the study outcomes and trade-off decision taken provide the evidence that justify the decisions. Note that the Justification Diagram is initialised during the step indicated by item 2. It can then be checked and updated based on the analyses of the study outcomes. The completed Justification Diagram forms a persisting record of why decisions were taken.

For one of the TOICA use-cases, Dassault Systèmes implemented a prototype Architects' Cockpit using their 3DEXPERIENCE[3] platform. CATIA RFLP [4] representation was used for modelling the aircraft architecture.

[3] `http://www.3ds.com`
[4] RFLP stands for Requirements, Functional, Logical, Physical.

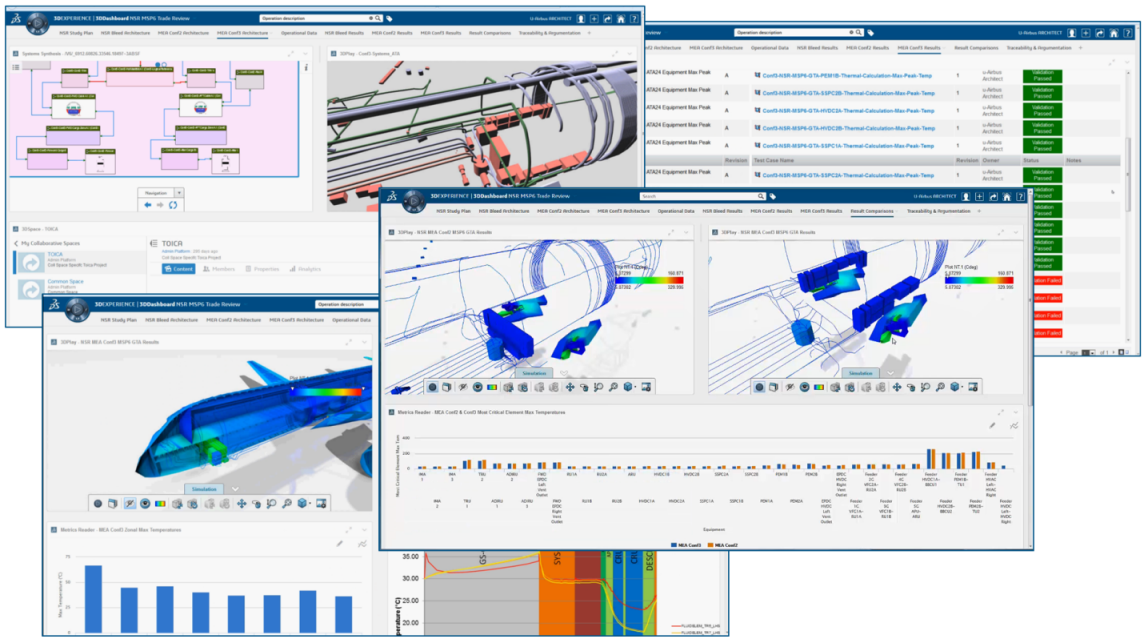

**Figure 2:** *Examples of architectures and study views provided by the architect's 3DDashboard*

Additional features, like sequence diagrams, were added in order to represent the operational scenarios describing the systems usage throughout a typical flight mission. Three configurations representing different architectural alternatives were modelled. This information formed the basis on which the studies were performed.

The study tasks (including references to inputs, objectives and expected results) are created by the architect and assigned to a various domain specialists, who would be working on their chosen platforms; these different platforms were inter-connected by the web-services [5].

When each element of the studies are completed, the resulting calculation and the key results are uploaded to the evolving collaboration database; again using web-services. When the discipline-specific platforms complete their respected data uploads, they send notifications to the Architects' Cockpit. When sufficient data has been computed (i.e. enough of the study steps have been competed) then the decision-making team can be convened to schedule a maturity review and evaluate the outcomes.

In order to support the collaborative review of the trade study, all the important information items (e.g. definition of alternative architectures, performance characteristics, key parameters, requirements status and lightweight representations of 3D results) are summarized in a multi-tab, widget-based modular environment called the *3DDashboard* (see Fig. 2). This example shows some of the many viewpoints provided by the Architects' Cockpit via the *3DDashboard*.

In Fig. 2, the top left-hand view shows a diagrammatic representation of the trade study. In its simplest form, it comprise interlinked block diagrams. The blocks represent the different steps of the trade study; the interconnections show the flow of data between the different steps of the study. Reading from left to right, this view shows the sequence in which the study will be performed. Note that each step may be allocated to a different discipline, within an organisation or in the extended enterprise. This inset also shows a 3D representation of the aircraft that the architects

[5]Web-services are a standardised way of integrating, and thus inter-operating, Web-based applications over an Internet protocol backbone

are looking to address. As [7] points out, multimodal explanations with 3D graphic technologies provides significant improve in knowledge integration amongst a varied audience of stakeholders. The top-right inset provides the architect with a status of the study step; green indicating that a study step has been completed and data entered into the collaboration database; amber indicating that it is being processed; and red indicating that it is yet to start.

The lower inset shows that the architect is able to view the results in multiple ways; e.g. as a colour-map superimposed on a 3D representation of the aircraft, or various graphs of the performance measures and key parameters. The foreground inset shows that the architect is able to zoom into regions that might require specific focus of attention. Our main aim was to remove the limitation of visualising the information on one screen. Therefore, we can view each inset on different screens; the key requirements being that the views must be synchronised and consistent.

This environment provides a collaborative decision support systems for the architect and their multi-disciplinary team. In this use-case, one of the "widgets" displayed a prototype of the Justification Diagram within a tab dedicated to traceability.

## 4.3. Decisions based on multi-disciplinary assessments of architectures

The TOICA team developed innovative methods for defining new aircraft architectures and designs [45]. These methods used various representations of the aircraft architecture to enable multi-disciplinary assessment of their characteristics through numerical simulations, multiple-criteria decision analysis and expert judgement. Our challenge was to make sense of the disparate analyses and then justify our decisions to accept or reject a particular design. In fact, it is very difficult to evaluate the soundness of the proposed architectures, because there are many sources of uncertainties arising from, for example, the models and the data. Further, for new architectures there may be incomplete data; the selection criteria will evolve as the understanding of the design evolves; and constraints on the design space that represents valued design will need to be discovered. As such, the Justification Diagram was selected to assess its applicability for recording the bases of having many decisions to take on whether or not to accept or reject a design. For a smaller subset of the problem, we created a Justification Diagram to facilitate the communication amongst the stakeholders and highlight the pertinent elements of the decisions. To identify the elements needed for a Justification Diagram, we interviewed specialists in various disciplines, including aircraft architects, and we studied 70 different documents (presentation supports, digital geometries, simulation documents, informal documentation and selection criteria stored as spreadsheets).

Note that, the Justification Diagram does not show the process; the process is accounted for in the study description. Nevertheless, the justification Diagram needs to be in the context of the study. Therefore, the justification Diagram should not be confused with behaviour diagram like UML [6] activity diagram or BPMN?[7]. The aim is to show the rationality and to build trust, to be confident, on the final result (the justification). So, in this diagram, sequence of activities is not described and input and output are in terms of rationality and not in terms of temporal process. We choose to construct the diagram from the perspective of an architect, and not at detailed level of a specialist; this diagram is intended for an architect to review and, in due course, to present to the chief engineer.

[6]Unified Modeling Language (http://www.uml.org/) maintained by the Object Management Group
[7]Business Process Modeling Notation (http://www.bpmn.org/) maintained by the Object Management Group

**Super integration**

As stated above, a full description of a product architecture should provide the perspectives needed for all the disciplines that interact to realise the product design. It is in this inclusive sense that the phrase *Super Integration* was introduced in the TOICA project; meaning to reason about product architectures as a multi-disciplinary whole, before it is elaborated or decomposed into its constituent parts. In this section, the terms *architecture* and *design* are used interchangeably because one person's architecture may be perceived as another person's design.

Specifically, *Super Integration* is an approach, supported by a system of integrated methods and principles, that enable product architects to seek valued architectures and designs. They do this by agilely manipulating the *functions*, and their relationships, the product requirements and the *physical allocations* of the functions. The Super Integration includes multiple systems thinking approaches, e.g. TRIZ [34], C-K Theory [28], and Systems Engineering [48]. In this context, the *functions* places focus on "what the product should, or should not, do", whilst the *physical allocations* focuses on the multiple ways that the functions may be implemented. The output of the Super Integration approach is the representation of a set of product architectures, both functional and physical, that provide a single source of information for further multi-disciplinary perspectives and analyses.

Further, by using the multi-disciplinary analyses, the product architects can narrow a set of possible designs to a set of feasible design. The main objective for narrowing the sets is to seek solutions that all stakeholders value. This objective is achieved by

- making inferences about their architectures using integrated multiple perspectives, comprising requirements, functions, operational logic and possible product elements
- manipulating the architectural elements to seek more desirable and valuable topologies, whilst maintaining coherency amongst the integrated multiple perspectives
- innovating product solutions by exploring highly dynamic allocation, re-allocation, association and re-associations amongst requirements, desired functions, intended operational logic and their allocation to product elements
- assessing the insertion of new product technologies as candidate solutions to implement their architectures
- seeking viability of their designs through multi-disciplinary analyses.

Set-based Design (SBD) is a process that allows parallel evaluation of multiple alternative solution concepts at the early stages of design. Conceiving and representing a sets of possible solutions initialises the set of possible design concepts. Then, by performing multi-disciplinary analyses, the set is narrowed down (infeasible members of the set are eliminated) and details of each design concept is increased. The narrowing of the set is done consensually amongst multi-disciplinary teams, which creates agreement and commitments from each of the disciplines. The key aspect of the SBD process is that a design is not selected, instead a sub-set of the infeasible designs are eliminated. This practice ensures that design options remain flexible for as long as possible and poorer performing options are only eliminated when sufficient knowledge has been acquired [17]. Essentially, Super Integration was conceived in TOICA as one way to initialise the set of *possible* solutions for a Set-Based Design process.

By using appropriate multi-disciplinary analyses, information is produced to acquire knowledge for "shelving", or eliminating, the least valued, inferior solutions (the set of *infeasible* solutions). The remaining solutions, forming a set of *feasible* solutions, are then analysed in more

detail to converge on a set of *competitive* solutions. The decision taken to shelve or accept a solution is also a key component of the acquired knowledge. Therefore, we need to record the justification for shelving, or accepting, each design solution such that the solutions may be re-used in a future design project. For example, a solution may have been shelved because we learnt that certain enabling technologies were not available at the right level of maturity, but in a future scenario the technology becomes available and the solutions may become feasible. If we did not know the initial reason for shelving the design, then we may not be able to assess whether that design could be feasible and valued in some future context.

This approach relies on a number of important features for the application of Super Integration to be sustainable:

a) rapid transition from a conceptual representation in one's mind, into representations for multi-disciplinary perspectives and analyses

b) appropriate analyses to learn fast for making informed decisions on whether to shelve or accept design solutions

c) recording the reasons for shelving or accepting design solution

d) at some future context to search and retrieve the shelved designs for reconsidering their feasibility

Only a superficial description of the Super Integration approach is given here. The fuller description of the Super Integration approach is outside the scope of this paper and will be the subject of another paper. This brief explanation is simply to establish that a persisting record that justifies a set of feasible solution is important because these early decisions already commit to a large part of the future project costs.

**Justification Diagram for alternative Environmental Control System**

As an illustrative example, consider that a product architect wants to assess an alternative solution for a cabin Environmental Control System (ECS) – a system that provides a comfortable environment for passengers and avionics equipment. Specifically, s(he) wants to start with a baseline comprising a current pneumatically powered engine bleed air system and replace it with a more electrically powered system (More Electric Aircraft - MEA). S(he) decides to represent it as a set of functions that the system must perform (functional representation), and a view of the topology of components that would implement these functions (logical representation). S(he) would like to base the decision on a number of *value drivers*, for example thermal performance, electrical and thermal behaviours as well as the resilience of the design to changes in any system component. These types of changes can be, for example, component redesign due to requirements changes during design, or a failure of a component during operations to force a transition from normal to abnormal behaviour.

In Fig. 3, we build the Justification Diagram that captures the information needed for the alternative solution for the ECS to be a feasible solution. The production of this Justification Diagram is a collaborative activity, where a representative of each impacted discipline participates in identifying the strategies, the sub-conclusions and the evidence to be gathered. At this stage, it is not known if the top-level conclusion will result in a positive or a negative outcome; the evidence will enable this to be determined in due course. The notation used is as discussed in Section 3 and summarised in Table 1. Typically a Justification Diagram is constructed from the top-level conclusions and elaborated to the set of evidence in a top-down manner; conversely, the

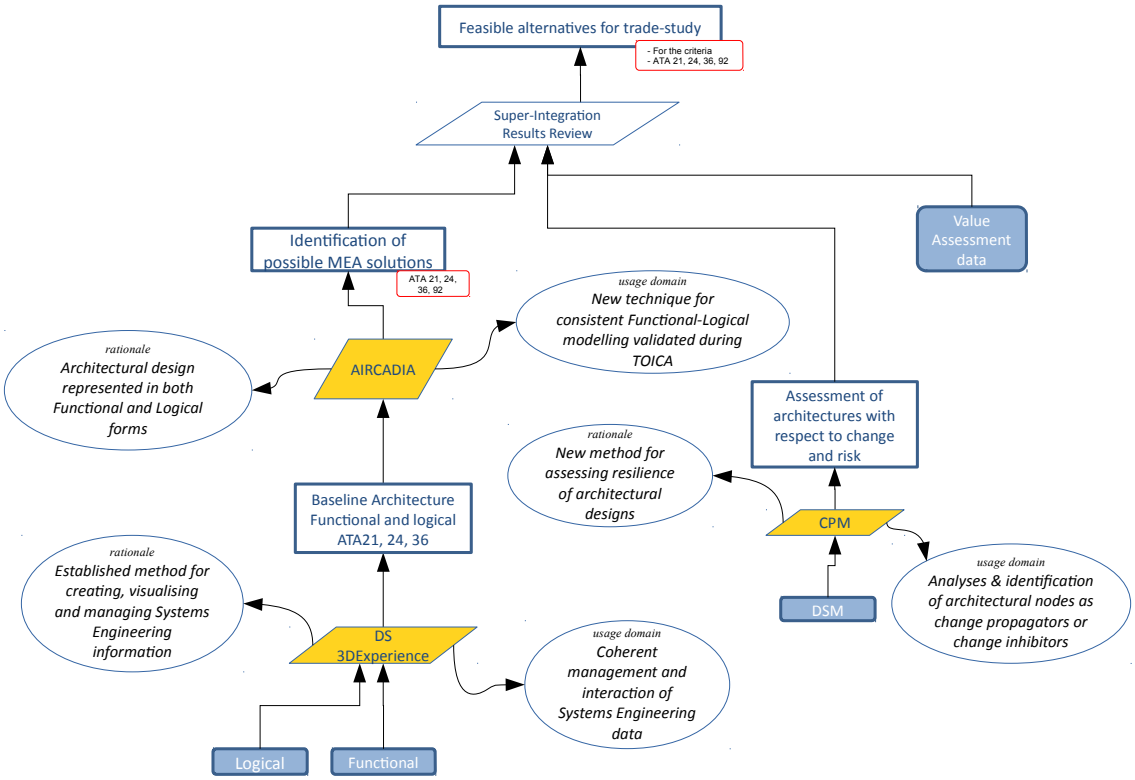


**Figure 3:** *Justification Diagram example: super integration*

information is populated from the evidence to justify the goal(s). This two-way approach allows the logic of the construct to tested, currently informally, as the information becomes available.

Suppose that the strategy is to perform a "Super-integration Results Review". One set of evidence identified is to reuse "Value Assessment data", which describes the figures of merits that will be used to assess the aptness of the solution. In order to quantify the figures of merits two conclusions are needed: "Identification of possible MEA solutions" and "Assessment of architectures with respect to change and risk".

Firstly, we focus on the left branch of the Justification Diagram. The software strategy proposed by the engineer is to use "AirCADia" because it allows architectural design to be represented in both functional and logical forms (the rationale). Such usage of AirCADia had been validated during the TOICA project (usage domain).

In order to use AirCADia, a conclusion based on an existing architecture is needed: "Baseline Architecture Functional and logical ATA21, 24, 36". According to the engineers, such a baseline exists, the "Logical" and "Functional" representations are managed by "DS 3DExperience" (the bottom left software strategy). The rationale and usage domain for DS 3DExperience are made explicit.

Secondly, we focus on the right branch of the Justification Diagram. The software strategy for assessing the robustness of the architecture to changes is selected to be CPM. Again, the rationale and usage domain are made explicit.

The conclusions arising from these two branches would enable the strategy "Super-integration Results Review" to be enacted.

Note that the Justification Diagram does not represent the process by which information is shared by the different software strategies. The management and provision of consistent information is the role of the Architects' Cockpit, as described in Section 4.2.

For populating the Justification Diagram, the product architect starts by describing the baseline architecture in terms of both Logical and Functional models. The information associated with both these representations needs to be created and managed, and as declared during the construction, s(he) uses Dassault Systèmes 3DExperience as her Architects' Cockpit (bottom left hand of the Justification Diagram). A Design Structure Matrix (DSM) provides a succinct representation of the elements of a system and their interactions [11]. This compact representation of the baseline architecture is used by the Change Propagation Method (CPM) [6] and for initialising AirCADia. Both CPM and AirCADia (software strategies) are used to establish the properties of the baseline architecture, against which the new electrical architecture will be assessed. The *rationale* and *usage domain* for each of these software strategies justifies their aptness for these outcomes.

AirCADia is also the software strategy for modifying the baseline architecture to a more electrically powered system. It allows the new MEA architecture of the ECS to be represented consistently in both Functional and Logical forms. AirCADia also produces a DSM representation of the MEA architecture, which is used by the CPM to generate its associated risk information. The combined output from both AirCADia and CPM form the Value Assessment Data, which is the evidence for the decisions made during the "Super-Integration Results Review".

During the TOICA research project, both AirCADia and CPM had been informally tested and agreed to be suitable for producing information to assess the key characteristics of aircraft systems architectures. AirCADia produces the information about the performance characteristics of the architecture; whilst CPM provides information about its resilience to change. Therefore, these *Software Strategies* were allocated to provide only the data for which they had been tested as appropriate sources. Similarly, 3DExperience had been developed as the prototype Architects' Cockpit and also tested to ensure that it was able to display the right information from the right sources. Thus, having established the justification approach, when the data from AirCADia and CPM is assessed, then the Justification Diagram is updated and the decision recorded.

To illustrate our approach further, we place a restriction that not all stakeholders' needs had been addressed. The two conclusions, "Identification of possible MEA solutions" and "Feasible alternatives for trade studies", have the constraint such that only one type of stakeholder has been addressed - the engineers. In this case they are aeronautics engineers representing Air Conditioning & Pressurization (ATA 21)[8], Electrical Power (ATA 24), Pneumatics (ATA 36) and Electrical System Installation (ATA 92).

**Global picture**

We give a simplified view of the Justification Diagram, without Usage domain and Justification boxes, for the global use case in Figure 4. Here, the final conclusion is that we have the best architectural choices. This conclusion is based, on the one hand, the fact that we have all the feasible architectures and, on the other hand, the results of analysis on criteria such as thermal or mass criteria are within the desired range. The strategy for moving to the conclusion is an architectural review (a committee of experts). It is a *human* strategy, and the strategy box must be linked to, for instance, the minutes of this meeting or a report of the committee's discussion [37].

Regarding the results of architecture analyses, on this diagram we have only included the thermal aspects. For this, we used the Global Thermal Aircraft (GTA) simulation. Naïvely, we can

[8]ATA chapters are defined by the *Air Transport Association of America*, it is a common referencing standard for all commercial aircraft documentation.

see the GTA simulation as an aggregation of simulations. The GTA results are key elements for decision-making such as choice of materials, identification of thermal hazards, qualification of equipment or architectural choices. There is a need to have strong confidence in the results of the GTA. Therefore, with thermal experts, we developed a Justification Diagram to justify confidence in these results. Thus, we have defined models of specific argumentation dedicated to the field of simulation. However, in the example given here, we show a diagram which deals with the choice of architecture and not on thermal simulations: the only interesting piece of information is to know that the strategy "GTA" simulation is validated. This is present in the justification of the strategy (not shown on the diagram for reasons of space) and this justification refers to the Justification Diagram of the GTA. In TOICA, we also produced a Justification Diagram for the GTA validation; however, this diagram is for the thermal expert and does not need to appear at the architect level.

Finally, the Justification Diagram is linked to 16 documents from the initial 70. For instance, an Excel file is linked to the evidence: "*Value Assessment data*". This excel file contains all the

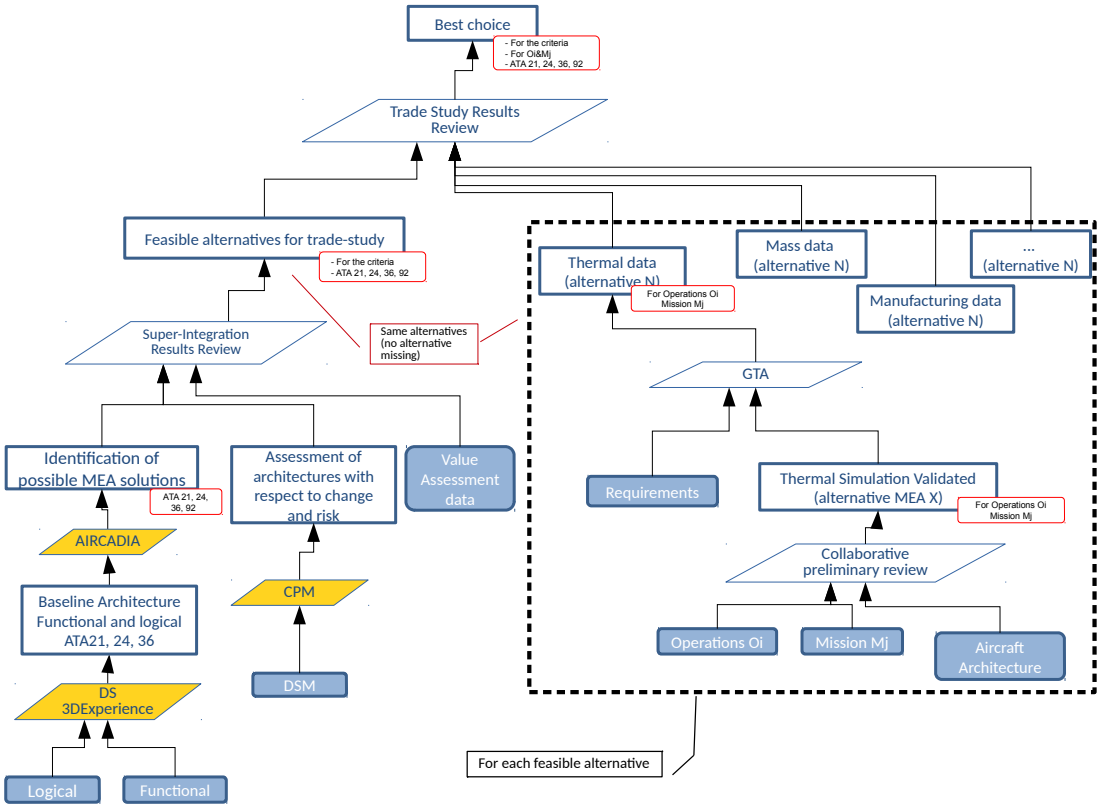


**Figure 4:** *Justification Diagram example: architect candidates*

information as captured from the stakeholders, in the form of expectations, needs, and then mapped to value drivers and value dimensions.

## 4.4. Implementation

In TOICA, we demonstrated how to use our Justification Diagram during a review of architectures in the architect's environment (Architects' Cockpit dashboard) and how it supports decision making. By showing key elements (mostly V&V elements), Justification Diagrams support the decision maker to understand why a result can be trusted and what are the weak points. In addition, Dassault Systèmes implemented a prototype version of the Justification Diagram capability in the 3DEXPERIENCE based Architects' Cockpit. This environment provides new opportunities with respect to the prevailing paper-based approach. By a simple click, it is possible to have access to all data and documents related to the diagram. This functionality increases confidence through a direct access to the justification information.

The difficulty in making this prototype lay mostly in the fact that the objects displayed in the Justification Diagram are of a very heterogeneous nature. Even if typical evidence elements like documents, models, simulation processes, methods, etc. can easily be matched to objects of the platform data-model, others like the conclusion do not have an obvious equivalent among the *product lifecycle management* objects. It was therefore decided to misuse a generic structure of requirements in 3DEXPERIENCE, in order to illustrate the feasibility and benefits of the concept. Requirement objects were convenient because they have a hierarchy that can be represented as a tree graph and references of platform objects can be attached to them.

For this TOICA use-case, we therefore populated that structure with the key evidence of the study and referenced it within one of the widgets of the 3DDashboard tab dedicated to traceability. It was shown at the end of the collaborative review to induce confidence and uphold the architect's decision.

## 4.5. Architect feedback

In the context of designing a complex system, such as an aircraft, requires managing many interactions between different components. It is mandatory to record rigorously the trade-off decisions and associated rationales. The formalisation and the traceability of the decisions rationales enable not only to make more robust the architecture design but also to enable work on design evolutions in case of late discovery of issues. The Justification Diagram is providing an efficient mean to address this architects' need.

The implementation of the Justification Diagram within all the studies contributing to the trade-off (thermal integration study in the frame of TOICA) provides trust in the study outcome. It also enables to highlight and to formalise the consequences of the uncertainties associated with some parameters. This is a key asset for architect for deciding about design margins to be considered for managing these uncertainties.

The integration of the Justification Diagram within the Architects' cockpit capabilities provides an efficient means to perform design reviews. Specifically, all the relevant information to make decision are presented in a synthetic way to the architect, with the ability to access additional information that might be required by the reviewers.

Another argument that must not be neglected, concerning the added value of the Justification Diagram, is the simplification of the V&V process. By applying this methodology at different level of design granularity and maturity, one can easily get the rationale on how the design addresses the targeted performances and requirements.

## 5. Conclusion

In this paper, after giving a brief reminder about Justification Diagram, we showed how we used this diagram in a true industrial use case. In TOICA European project, it has been shown that the Justification Diagram supports the architects and stakeholders to understand the rationale of V&V activities. Indeed, such diagram, when it is include in software like the architect cockpit, allows to navigate through the V&V information and to find areas where there is a lack of evidence.

In addition, one aim of using Justification Diagrams is to cope with what we call the tsunami of documents. Today, every process produces many documents and experts and architects need to make sense of it. By giving a global picture, a graphical representation of the rational, the Justification Diagram could support the architect to deal with the tsunami of V&V documents. The drive towards digitalisation is creating an imperative to manage information, and not documents; the Justification Diagrams is immediately applicable to both these approaches, indeed to a heterogeneous mix of these approaches.

To conclude, the Justification Diagram also fits into the notions of modularity; it supports V&V tasks for reusability of a simulation or a simulation product: if a method, the inputs or the evidence change, then it is easier to understand the impact on the overall V&V arguments.

## References

[1] Association for the Advancement of Artificial Intelligence. *Proceedings of the the AAAI Spring Symposium on producing cooperative explanations*, 1992.

[2] Osman Balci. Verification, validation, and accreditation. In *Proceedings of the 30th conference on Winter simulation*, pages 41–4. IEEE Computer Society Press, 1998.

[3] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2008.

[4] Valentin Cassano and T. S. E. Maibaum. The definition and assessment of a safety argument. In *25th IEEE International Symposium on Software Reliability Engineering Workshops, ISSRE Workshops*, pages 180–185. IEEE Computer Society, 2014.

[5] B. Chandrasekaran, Michael C. Tanner, and John R. Josephson. Explaining control strategies in problem solving. *IEEE Intelligent Systems*, 4:9–15, 19–24, 1989.

[6] P. John Clarkson, Caroline Simons, and Claudia Eckert. Predicting change propagation in complex design. *Journal of Mechanical Design*, 126(5):788–797, October 2004.

[7] Brent H. Daniel, William H. Bares, Charles B. Callaway, and James C. Lester. Student-sensitive multimodal explanation generation for 3d learning environments. In Jim Hendler and Devika Subramanian, editors, *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA.*, pages 114–120. AAAI Press / The MIT Press, 1999.

[8] Defence Standard 00-56 Safety Management Requirements for Defence Systems Part 1 : Requirements, p9, section 9.1. Standard, UK Ministry of Defence, 2007.

[9] Clément Duffau, Thomas Polacsek, and Mireille Blay-Fornarino. Support of justification elicitation: Two industrial reports. In *Advanced Information Systems Engineering - 30th International Conference, CAiSE 2018, Tallinn, Estonia, 2018, Proceedings*, Lecture Notes in Computer Science, pages 71–86. Springer, 2018.

[10] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[11] Steven D. Eppinger and Tyson R. Browning. *Design Structure Matrix Methods and Applications*. The MIT Press, 2016.

[12] M. Sinan Gönül, Dilek Önkal-Atay, and Michael Lawrence. The effects of structural characteristics of explanations on use of a DSS. *Decision Support Systems*, 42(3):1481–1493, 2006.

[13] Bryce Goodman and Seth R. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[14] Jean Goodwin and Alec Fisher. Wigmore's Chart Method. *Informal Logic*, 20(3):223–243, 2001.

[15] Thomas F. Gordon and Douglas Walton. The carneades argumentation framework - using presumptions and exceptions to model critical questions. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *COMMA*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 195–207. IOS Press, 2006.

[16] Reyes Grangel, Ricardo Chalmeta, and Cristina Campos. A modelling framework for sharing knowledge. In Bruno Apolloni, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007. Proceedings, Part II*, volume 4693 of *Lecture Notes in Computer Science*, pages 1230–1237. Springer, 2007.

[17] Alexander W. Gray, Douglas T. Rigterink, and Peter McCauley. Point-based versus set-based design method for robust ship design. *Naval Engineers Journal*, 129(2):83–96, June 2017.

[18] Allison Rose Greenwald. *Learning how to argue: experiences teaching the Toulmin model to composition students*. PhD thesis, Iowa State University, 2007.

[19] Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23(4):497–530, 1999.

[20] GSN Community. GSN Community Standard Version 1, 2011.

[21] David Harel. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program.*, 8(3):231–274, 1987.

[22] Maralee Harrell and Danielle Wetzel. *Using Argument Diagramming to Teach Critical Thinking in a First-Year Writing Course*, pages 213–232. Palgrave Macmillan US, New York, 2015.

[23] International Joint Conferences on Artificial Intelligence. *Proceedings of the IJCAI'93 Workshop on Explanation and Problem Solving*, 1993.

[24] Systems and software engineering – Systems and software assurance – Part 2: Assurance case. Standard, International Organization for Standardization, 2011.

[25] Tim Kelly and Rob Weaver. The goal structuring notation /- a safety argument notation. In *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004.

[26] Werner Kunz and Horst Rittel. Issues as elements of information systems. Working Paper 131, Institute of Urban and Regional Development, University of California, Berkeley, California, 1970.

[27] M. Lawrence, L. Davies, M. O'Connor, and P. Goodwin. Improving forecast utilization by providing explanations. In *21st International Symposium on Forecasting*, pages 9–24, 2001.

[28] Pascal Le Masson, Benoit Weil, and Armand Hatchuel. *Designing in an Innovative Design Regime—Introduction to C-K Design Theory*, pages 125–185. Springer International Publishing, Cham, 2017.

[29] Allan MacLean, Richard M. Young, Victoria M. E. Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. *Hum.-Comput. Interact.*, 6(3):201–250, September 1991.

[30] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[31] Hugo Mercier and Dan Sperber. Why do humans reason? arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34(2):57âĂŞ74, 2011.

[32] Ingrid Nunes, Simon Miles, Michael Luck, Simone Diniz Junqueira Barbosa, and Carlos José Pereira de Lucena. Pattern-based explanation for automated decisions. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 669–674. IOS Press, 2014.

[33] OMG. Business Process Model and Notation (BPMN), Version 2.0, 2011.

[34] Michael A. Orloff. *Toward the Modern TRIZ*, chapter 2, pages 19–30. Springer International Publishing, Cham, 2017.

[35] Chaim Perelman and Lucie Olbrechts-Tyteca. *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press, June 1969.

[36] Thomas Polacsek. Validation, accreditation or certification: a new kind of diagram to provide confidence. In *10th IEEE International Conference on Research Challenges in Information Science, RCIS,*, pages 59–466, 2016.

[37] Thomas Polacsek and Laurence Cholvy. A framework to report and to analyse a debate. In Weiming Shen, Jean-Paul A. Barthès, Junzhou Luo, Peter G. Kropf, Michel Pouly, Jianming Yong, Yunjiao Xue, and Milton Pires Ramos, editors, *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2011, June 8-10, 2011, Lausanne, Switzerland*, pages 84–90. IEEE, 2011.

[38] Thomas Polacsek, Stéphanie Roussel, François Bouissiere, Claude Cuiller, Pierre-Eric Dereux, and Stéphane Kersuzan. Towards thinking manufacturing and design together: An aeronautical case study. In Heinrich C. Mayr, Giancarlo Guizzardi, Hui Ma, and Oscar Pastor, editors, *Conceptual Modeling - 36th International Conference, ER 2017, Valencia, Spain, November 6-9, 2017, Proceedings*, volume 10650 of *Lecture Notes in Computer Science*, pages 340–353. Springer, 2017.

[39] Henry Prakken. Formal systems for persuasion dialogue. *Knowledge Eng. Review*, 21(2):163–188, 2006.

[40] Iyad Rahwan. Mass argumentation and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1), 2008.

[41] Chris Reed, Douglas Walton, and Fabrizio Macagno. Argument diagramming in logic, law and artificial intelligence. *Knowl. Eng. Rev.*, 22:87–109, 2007.

[42] G. W. A. Rowe, C. A. Reed, and J. Katzav. Araucaria: Marking up argument. In *Working Notes of the European Conference on Computing and Philosophy*, Glasgow, Scotland, 2003.

[43] Glenn Rowe and Chris Reed. Argument diagramming: The araucaria project. In Lakhmi Jain, Xindong Wu, Tony Sherborne, Simon J. Buckingham Shum, and Alexandra Okada, editors, *Knowledge Cartography*, Advanced Information and Knowledge Processing, pages 163–181. Springer London, 2008.

[44] James E. Rumbaugh, Ivar Jacobson, and Grady Booch. *The unified modeling language reference manual*. Addison-Wesley-Longman, 1999.

[45] Sanjiv Sharma, Christoffer Levandowski, Arturo Molina-Cristobal, Timoleon Kipouros, Ola Isaksson, and Trevor Robinson. Super Integration: seeking novel valued solutions. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016)*, 2016.

[46] Richard W. Southwick. Explaining reasoning: an overview of explanation in knowledge-based systems. *Knowledge Eng. Review*, 6(1):1–19, 1991.

[47] William R. Swartout. XPLAIN: A system for creating and explaining expert consulting programs. *Artif. Intell.*, 21(3):285–325, 1983.

[48] Technical Committee: ISO/IEC JTC 1/SC 7. Systems and software engineering – system life cycle processes, 2015.

[49] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, Cambridge, UK, 2003. Updated Edition, first published in 1958.

[50] USA Department of Defense (DoD). DoD Modeling and Simulation (M&S) glossary (DoD 5000.59-m), 2007.

[51] Axel van Lamsweerde. *Requirements Engineering - From System Goals to UML Models to Software Specifications.* Wiley, 2009.

[52] Bart Verheij. Argumentation support software: boxes-and-arrows and beyond. *Law, Probability and Risk*, 6(1-4):187–208, 2007.

[53] Bart Verheij. The toulmin argument model in artificial intelligence. In Guillermo Ricardo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 219–238. Springer, 2009.

[54] Douglas Walton. Visualization tools, argumentation schemes and expert opinion evidence in law. *Law, Probability and Risk*, 6(1-4):119–140, 2007.

[55] Bin Wei and Henry Prakken. An analysis of critical-link semantics with variable degrees of justification. *Argument & Computation*, 7(1):35–53, 2016.

[56] Ursula Wingate. 'argument!' helping students understand what essay writing is about. *Journal of English for Academic Purposes*, 11(2):145–154, 6 2012.

[57] L Richard Ye. The value of explanation in expert systems for auditing: An experimental investigation. *Expert Systems with Applications*, 9(4):543–556, 1995.

[58] L. Richard Ye and Paul E. Johnson. The impact of explanation facilities in user acceptance of expert system advice. *MIS Quarterly*, 19(2):157–172, 1995.