Gap analysis of existing reasoning techniques and requirements for the ATM participatory architectural design

D4.1
PACAS
Grant: 699306
Call: H2020-SESAR-2015-1
Topic: Sesar-10-2015 ATM Architecture
Consortium coordinator: UNITN
Edition date: 22 December 2016
Edition: 00.02.00
**Authoring & Approval**

### Authors of the document

<table>
<thead>
<tr>
<th>Name/Beneficiary</th>
<th>Position/Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paolo Giorgini/UNITN</td>
<td>Project Coordinator</td>
</tr>
<tr>
<td>Elda Paja/UNITN</td>
<td>Project member</td>
</tr>
<tr>
<td>Roberta Cuel/UNITN</td>
<td>Project member</td>
</tr>
<tr>
<td>Diego Ponte/UNITN</td>
<td>Project member</td>
</tr>
<tr>
<td>Giusi Orabona/UNITN</td>
<td>Project member</td>
</tr>
<tr>
<td>Børge Haugset/SINTEF</td>
<td>Project member</td>
</tr>
<tr>
<td>Robert Bains/SINTEF</td>
<td>Project member</td>
</tr>
<tr>
<td>Narve Lyngby/SINTEF</td>
<td>Project member</td>
</tr>
<tr>
<td>Ola Løkberg/SINTEF</td>
<td>Project member</td>
</tr>
<tr>
<td>Erlend Andreas Gjære/SINTEF</td>
<td>WP3 Leader</td>
</tr>
</tbody>
</table>

### Reviewers internal to the project

<table>
<thead>
<tr>
<th>Name/Beneficiary</th>
<th>Position/Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabiano Dalpiaz/UUTR</td>
<td>WP2 Leader</td>
<td>30/07/2016</td>
</tr>
<tr>
<td>Elisa Chiarani/UNITN</td>
<td>Project Assistant</td>
<td>04/08/2016</td>
</tr>
<tr>
<td>Fabiano Dalpiaz/UUTR</td>
<td>WP2 Leader</td>
<td>25/08/2016</td>
</tr>
<tr>
<td>Elisa Chiarani/UNITN</td>
<td>Project Assistant</td>
<td>25/08/2016</td>
</tr>
</tbody>
</table>

### Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

<table>
<thead>
<tr>
<th>Name/Beneficiary</th>
<th>Position/Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paolo Giorgini</td>
<td>Project coordinator</td>
<td>31/08/2016</td>
</tr>
<tr>
<td>Paolo Giorgini</td>
<td>Project coordinator</td>
<td>22/12/2016</td>
</tr>
</tbody>
</table>

### Rejected By - Representatives of beneficiaries involved in the project

<table>
<thead>
<tr>
<th>Name/Beneficiary</th>
<th>Position/Title</th>
<th>Date</th>
</tr>
</thead>
</table>
## Document History

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
<th>Status</th>
<th>Author</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.00.01</td>
<td>13/07/2016</td>
<td>Draft</td>
<td>Elda Paja</td>
<td>Proposed table of contents</td>
</tr>
<tr>
<td>00.00.02</td>
<td>20/07/2016</td>
<td>Agreed structure</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>After discussion</td>
</tr>
<tr>
<td>00.00.03</td>
<td>25/07/2016</td>
<td>Added sections</td>
<td>Elda Paja</td>
<td>Added section on reasoning for security, and the one on multi-objective trade-off analysis</td>
</tr>
<tr>
<td>00.00.04</td>
<td>29/07/2016</td>
<td>Added sections</td>
<td>Roberta Cuel, Diego Ponte</td>
<td>Added section on reasoning for economics and organizational perspectives</td>
</tr>
<tr>
<td>00.00.05</td>
<td>29/07/2016</td>
<td>Completed 1st draft of section</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Sections 1.4.2 and 1.5</td>
</tr>
<tr>
<td>00.00.06</td>
<td>30/07/2016</td>
<td>Scientific Review</td>
<td>Fabiano Dalpiaz</td>
<td>Scientific review completed</td>
</tr>
<tr>
<td>00.00.07</td>
<td>04/08/2016</td>
<td>Quality check</td>
<td>Elisa Chiarani</td>
<td>Quality check completed. Minor improvements</td>
</tr>
<tr>
<td>00.00.08</td>
<td>08/08/2016</td>
<td>Restructuring</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Restructured deliverable to address reviewer’s comments</td>
</tr>
<tr>
<td>00.00.09</td>
<td>11/08/2016</td>
<td>Added sections</td>
<td>Elda Paja</td>
<td>Added material on reasoning for economics and organizational perspectives as intended in PACAS</td>
</tr>
<tr>
<td>00.00.10</td>
<td>19/08/2016</td>
<td>Added sections</td>
<td>Erlend Andreas Gjære, Børge Haugset, Robert Bains, Narve Lyngby, Ola Løkberg</td>
<td>Added material on safety reasoning and needs from past experience.</td>
</tr>
<tr>
<td>00.00.11</td>
<td>22/08/2016</td>
<td>Revised sections</td>
<td>Roberta Cuel, Diego Ponte, Giusi Orabona</td>
<td>Review of the economic and organizational parts, review of Section 2.2.2 and 2.2.3, and sections 3.1 and 3.2.</td>
</tr>
<tr>
<td>00.00.12</td>
<td>23/08/2016</td>
<td>Revised and added sections</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Review of the economic and organizational parts, review of section on</td>
</tr>
<tr>
<td>Date</td>
<td>Action</td>
<td>Revised by</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------</td>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>23/08/16</td>
<td>Revised sections</td>
<td>Roberta Cuel, Diego Ponte, Giusi Orabona</td>
<td>Review of the SWOT analysis description, revised MADA in section 2.2., revised section 3.3</td>
<td></td>
</tr>
<tr>
<td>24/08/16</td>
<td>Revised sections; Added sections</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Merged section 2.2.3 with section 2.2.2. Added section 4 on requirements and section 5.</td>
<td></td>
</tr>
<tr>
<td>25/08/16</td>
<td>Revised section</td>
<td>Erlend Andreas Gjære</td>
<td>Added requirements in section 4 based on section 3.3</td>
<td></td>
</tr>
<tr>
<td>25/08/16</td>
<td>Revised section; Made Elda Paja a complete pass on the deliverable</td>
<td>Elisa Chiarani</td>
<td>Revised section 4, added description for the new requirements.</td>
<td></td>
</tr>
<tr>
<td>25/08/16</td>
<td>Quality check</td>
<td>Elisa Chiarani</td>
<td>Minor changes</td>
<td></td>
</tr>
<tr>
<td>26/08/16</td>
<td>Minor improvements</td>
<td>Diego Ponte, Roberta Cuel</td>
<td>Addressed reviewer’s comments on economic and organizational section.</td>
<td></td>
</tr>
<tr>
<td>29/08/16</td>
<td>Revised document</td>
<td>Elda Paja</td>
<td>Addressed reviewer’s comments, applied quality check’s changes.</td>
<td></td>
</tr>
<tr>
<td>30/08/16</td>
<td>Revised section</td>
<td>Erlend Andreas Gjære</td>
<td>Revised section 2.3 intro.</td>
<td></td>
</tr>
<tr>
<td>30/08/16</td>
<td>Revised section</td>
<td>Elda Paja</td>
<td>Revised section 2.2.2.</td>
<td></td>
</tr>
<tr>
<td>30/08/16</td>
<td>Revised section</td>
<td>Erlend Andreas Gjære</td>
<td>Revised 2.3.1 and 4.</td>
<td></td>
</tr>
<tr>
<td>31/08/16</td>
<td>Revised section</td>
<td>Roberta Cuel</td>
<td>Revised section 2.1.</td>
<td></td>
</tr>
<tr>
<td>31/08/16</td>
<td>Revised section</td>
<td>Elda Paja</td>
<td>Revised section 2.2.2 and section 4.</td>
<td></td>
</tr>
<tr>
<td>31/08/16</td>
<td>Final version</td>
<td>Elisa Chiarani</td>
<td>Final check</td>
<td></td>
</tr>
<tr>
<td>31/08/16</td>
<td>Final version</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Final document ready for approval and submission.</td>
<td></td>
</tr>
<tr>
<td>09/11/16</td>
<td>Added section</td>
<td>Fabiano Dalpiaz</td>
<td>Added section on connection with knowledge structures in ATM, e.g. AIRM.</td>
<td></td>
</tr>
<tr>
<td>19/12/16</td>
<td>Revised sections</td>
<td>Per Håkon Meland</td>
<td>Added requirement on connection with EATMA.</td>
<td></td>
</tr>
</tbody>
</table>
### D4.1 GAP ANALYSIS OF EXISTING REASONING TECHNIQUES AND REQUIREMENTS FOR THE ATM PARTICIPATORY ARCHITECTURAL DESIGN

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Type</th>
<th>Author(s)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.01.03</td>
<td>19/12/2016</td>
<td>Revised document</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Revised document structure, revised document for consistency.</td>
</tr>
<tr>
<td>00.01.04</td>
<td>21/12/2016</td>
<td>Quality check</td>
<td>Elisa Chiarani</td>
<td>Minor changes</td>
</tr>
<tr>
<td>00.02.00</td>
<td>22/12/2016</td>
<td>Final document</td>
<td>Elda Paja, Paolo Giorgini</td>
<td>Deliverable ready for submission.</td>
</tr>
</tbody>
</table>
Abstract

The main objective of PACAS WP4 is to analyse existing literature on methods, techniques and tools for multi-view model verification and multi-objective reasoning. In this deliverable, we identify the gap between the domain stakeholders’ needs (from the PACAS advisory board) and the state of art reasoning techniques identified from our literature review. The identification of gaps is complemented by an understanding of the needs from past experience in the ATM domain. As a result, the outcome of this deliverable is a gap analysis finalized with the identification of requirements for the PACAS reasoning support.
Table of Contents

Abstract ........................................................................................................................................... 6

Executive summary ......................................................................................................................... 8

1 Introduction ................................................................................................................................. 9
  1.1 PACAS Overview .................................................................................................................. 9
  1.2 Relationship with other deliverables .................................................................................. 10
  1.3 Structure of this document .................................................................................................. 10

2 State of the Art ............................................................................................................................ 11
  2.1 Economic and organizational reasoning techniques ......................................................... 11
    2.1.1 Human-centric reasoning ............................................................................................ 11
    2.1.2 Semi-automated and automated reasoning with BSc and SWOT .............................. 16
    2.1.3 Conceptual modelling approaches to economic and organizational reasoning .. 18
  2.2 Security reasoning techniques ............................................................................................. 19
    2.2.1 Socio-technical security reasoning techniques .......................................................... 19
    2.2.2 International security standards ................................................................................ 20
  2.3 Safety reasoning techniques ............................................................................................... 22
    2.3.1 Safety reasoning regulated by CENELEC standards ................................................ 23
    2.3.2 Safety techniques ....................................................................................................... 26
    2.3.3 Safety reasoning regulated by COMMISSION IMPLEMENTING REGULATION (EU) No 402/2013 ... 27
    2.3.4 Safety reasoning beyond standards ............................................................................. 28
  2.4 Multi-view multi-objective reasoning techniques .............................................................. 29
    2.4.1 Reasoning for conflict identification ......................................................................... 29
    2.4.2 Finding trade-offs among multiple objectives for decision-making ...................... 31

3 Identification of the needs ......................................................................................................... 33
  3.1 Interview method .................................................................................................................. 33
  3.2 Results from interviews and Validation Workshop #1 ..................................................... 33
  3.3 Needs from experiences on reasoning in the ATM domain ............................................ 34
    3.3.1 Identifying similarity and relatedness ....................................................................... 34
    3.3.2 Past experiences ........................................................................................................ 36

4 Requirements for the PACAS reasoning techniques ............................................................... 39

5 Conclusions ............................................................................................................................... 42

References ..................................................................................................................................... 43
Executive summary

PACAS develops and promotes the adoption of an innovative participatory change management process in complex socio-technical systems, such as Air Traffic Management (ATM) systems.

This deliverable serves two major objectives: (i) to consolidate the state of the art on reasoning within each PACAS perspective and that on multi-view multi-objective reasoning, and (ii) to identify the requirements for the reasoning techniques as a result of an analysis of the gaps that exist either from the needs identified from past experience in the field or from the needs that have been elicited through interviews and interactions with stakeholders from the ATM domain.

Our study reports on several aspects of reasoning techniques, with a special focus on their use in ATM. As such, we start our analysis by taking into account the modelling languages and frameworks reported in D3.1, and further investigate those that offer reasoning support. This initial analysis is then expanded with other dedicated reasoning procedures. Hence, we report on reasoning techniques specifically tailored for each of the PACAS stakeholder perspectives including economic and organizational, safety and security views, to then analyse multi-view and multi-objective reasoning techniques for change management.

The needs from the stakeholders have been gathered by following the first version of the identification method that is developed in D5.1. For the purposes of this deliverable, the identification was conducted through interviews with members of the PACAS advisory board and through interactions during the 1st validation workshop in Rome. Furthermore, the findings have been complemented with knowledge from past expertise in the ATM domain of the researchers of the PACAS consortium.

Our study identifies a set of preliminary requirements for the PACAS reasoning techniques, which are at the basis of the PACAS decision-making process. These requirements either derive from needs obtained from the interviews, are gaps in the literature that are relevant for PACAS, are suggestions of best practices from the state-of-the-art and needs from past experience, or derive from the partners’ expertise in the field. It is however important to emphasize that although we have identified needs related to the experience on reasoning in the ATM domain, our findings at this point are based and validated with a narrow sample of stakeholders (only few of the advisory board members provided input on the matter), and more investigation is needed to elaborate the requirements through iterative development of reasoning techniques and hypotheses to be explored by all the domain experts part of the PACAS Advisory Board.
1 Introduction

1.1 PACAS Overview

ATM systems are complex systems-of-systems that are managed via a layered architectural model, which includes operational, organisational, and technical layers to ease handling complexity. Due to strong interdependencies in an ATM system, any change introduced in any of these layers might trigger changes both within the same layer and in the other layers. Understanding all possible consequences of a design decision in ATM systems is a challenge due to the complexity of these systems and the existence of tight interdependencies within the ATM architecture. A careful consideration of possible changes together with their implications on the entire ATM system is crucial to support decision-making, while making sure that the ATM system does not suffer from any issues with respect to functionality, safety, security, performance, cost efficiency, or other desired characteristics of a well-functioning ATM system.

Figure 1: Multi-view multi-objective gamified participatory design process for ATM architectural change management

PACAS is about supporting change management in ATM systems from an architectural point of view, relying on the end-to-end inclusion of ATM domain stakeholders through gamification. The project constructs a platform that facilitates understanding, modelling and analysis of changes in the ATM system at different layers of abstraction. The approach to finding optimal solutions is based on a novel participatory design process to handle change management. The process relies on the provision of multiple views (to accommodate the expertise of the various domain stakeholders), as
well as the representation and analysis of multiple objectives, namely those related to economical, organizational, security, and safety concerns (Figure 1).

1.2 Relationship with other deliverables

This document relates to the following deliverables:

D2.1: the use of reasoning techniques is tightly related to the overall PACAS participatory gamified process. The results of the reasoning will be important input for the message exchange and alerting end users in order to deal with changes.

D2.2: this deliverable proceeds in parallel on the gap analysis of the reasoning techniques to be supported by the PACAS platform.

D3.1: the reasoning techniques will be affected by the modelling concepts identified by the gap analysis performed in D3.1.

D5.1: the reasoning techniques described in this deliverable can be applied in the scenario described in D5.1. At the same time, the toy example presented in detail in D2.2 is used to identify gaps related to reasoning techniques from the PACAS Advisory Board.

1.3 Structure of this document

This document is structured as follows.

Section 2 presents state of art techniques for reasoning within each perspective to be supported by PACAS, that is economic, organizational, security and safety, as well as techniques to reason across perspectives in order to find a trade-off among the various perspectives, requiring to find a trade-off among multiple objectives holding simultaneously.

Section 3 reports on the activities that were conducted to identify the needs with respect to reasoning techniques to be supported in the project. This is done by representing both needs from past experience of project members as well as needs that were identified through interviews with the advisory board members.

Finally, Section 4 analyses the gaps between the state-of-the-art in the field and the identified needs towards a set of requirements for the automated reasoning.

Finally, Section 5 concludes the deliverable.
2 State of the Art

This chapter reviews and consolidates the state-of-the-art literature on reasoning techniques for the PACAS platform. This is an important initial step towards eliciting the requirements for the PACAS reasoning techniques. We first focus on reasoning techniques for each of the four PACAS views, that is, how we can reason in terms of economic factors and in terms of organizational aspects, security problems, and safety concerns respectively. Then we discuss multi-view multi-objective reasoning techniques.

2.1 Economic and organizational reasoning techniques

We review the reasoning support for economic and organizational models. As already stated in D3.1, economic, business and organizational topics are intertwined and used together for the development of several models, with the most known being the 5 Forces model, the Value Chain, the SWOT model, the Strategy Maps and the Balanced Scorecard, the organizational model of Mintzberg. Here, we present reasoning techniques involving the Balanced Scorecards and the SWOT matrix (Section 2.1.1). It is worth noting that these models are considered very helpful in organization science and management to make strategic decisions, but the knowledge represented in them greatly depends on the human interpretation of the scenario, the set of priorities the decision makers consider relevant, and a set of quantitative elements (KPIs) that may help decision makers to make a choice. Since the most used analysis techniques are based on tacit and non-formalized knowledge, it is very hard and sometimes inconvenient to formalize them in an automated or semi-automated fashion to be integrated in support systems like the PACAS platform. On the other hand, from the 1st PACAS validation workshop it emerged that decision makers are very interested to improve their decisions using also semi-automated or automated reasoning techniques.

Therefore, we first discuss human-centric approaches in Section 2.1.1, to then investigate how to provide support for decision makers in terms of semi-automated or automated reasoning techniques for Balanced Scorecards and SWOT in Section 2.1.2. Finally, we deepen our discussion going over software engineering approaches that aim at representing and reasoning about economic and business-related concepts (Section 0), given the need to create reasoning techniques that will be integrated in the PACAS platform.

2.1.1 Human-centric reasoning

As depicted in literature, multi-view reasoning and decision-making are very complex to model because several factors come into play, such as:
1. the cognitive ability of individuals involved;
2. the relationships between the individuals involved;
3. the set of incentives which affect intrinsic and extrinsic motivations;
4. the organizational priorities;
5. environmental contingencies (laws, norms, etc.).

Although any decision process is biased by epistemological assumptions of human being, behaviours, organizational culture, economic approaches, etc., a set of factors and properties are commonly shared in decision-making models. Referring to Little [1], a decision model must be:

1. Simple. Simplicity and level of abstraction promote an ease understanding. Important phenomena should be modelled and the less important should be left out. Researchers are often pushed to add many details into a model, but this reduces its representativeness. This should be resisted, until the users demonstrate they are ready to assimilate it.

2. Robust/Reliable. User should perceive the model correct, with no bad answers. In other words, the model provides always meaningful range of values.

3. Easy to control. A user should be able to make the model behave the way he wants it to. For example, he should know how to set inputs to get almost any outputs. This seems to suggest that the user could have a preconceived set of answers and simply fudge the inputs until he gets them. Wherever objective accuracy is attainable, the vast majority of managers will seize it eagerly. Where it is not, which is most of the time, managers should be left in control. Thus, the goal of parameterization is to represent the operation as the manager sees it. We believe that if the managers cannot control the model they will not use it because of they fear it will coerce them to do something they do not believe in.

4. Adaptive. The model should be capable of being updated as new information, principles, rules, etc. becomes available. This is especially true for the parameters and the reasoning structure too.

5. Complete on important issues. Completeness is in conflict with simplicity. Structures must be found that can handle many phenomena without boding down. An important aid to completeness is the incorporation of subjective judgments. People have a way of making better decisions than their data seem to warrant. It is clear that they are able to process a variety of inputs and come up with aggregate judgments about them. Many, if not most, of the big advances in scientific knowledge come from measurement. Nevertheless, at any given point in time, subjective estimates will be valuable for quantities that are currently difficult to measure or which cannot be measured in the time available before a decision must be made. One problem posed by the use of subjective inputs is that they personalize the model to the individual or group that makes the judgments. This makes the model, at least superficially, more fragile and less to be trusted by others than a totally empirical model. However, the model with subjective estimations may be good enough, because it is more complete and coherent with a more realistic scenario.
6. **Easy to communicate with.** The manager should be able to change inputs easily and obtain outputs quickly.

Scholars agree that decisions with very important consequences for an entire system (e.g., an organization) are frequently made by groups [2]–[4]. This is due to the fact that the essence of each organization is to coordinate diverse persons, activities and functions, and its objective is to accomplish a goal that cannot be achieved by any of the group members working alone [5]. Therefore, any reasoning activity carried on in the organization strongly depends on human perspectives. These should be very informal relying mainly on human behaviour, and not formalized knowledge. Among others we have chosen 4 approaches: brainstorming, 7s McKinsey, SWOT and scenario analysis.

**Brainstorming**

This is a technique by which efforts are made to find a solution for a specific problem by gathering a list of ideas and comparing various views. Instead of formalizing the different views in a model, individuals are called to share the same space and time sharing their own opinion and knowledge, finding a common vision and view on the problems. Brainstorming is based on four fundamental principles:

1. Criticism is not allowed in a brainstorming session. Discriminatory remarks concerning the ideas of others are forbidden. In considering the ideas of others, one can understand the different perspective focusing on chains of mutual associations which emerge from the discussion.

2. Participants should mention all ideas coming into their minds without any constraint. Even crazy and unrealistic ideas may contain an element essential to the ultimate solution or initiate the association of realistic propositions in the minds of other members.

3. The more that ideas are produced within a relatively short time, the greater the probability that there is a really good and common one.

4. The function of the chairman is critical because she should establish a free and open atmosphere and not permit anyone to be suppressed. This requires very sensitive chairmanship [6].

**7S McKinsey**

The 7S McKinsey model is a Value Based Management (VBM) model that is intended to provide a company with a framework with the intent of generating value within the organization. The model considers the company organization as a mix of 6 dimensions that function around the seventh one, i.e. the Shared Values of a Company [7]. The six dimensions are: Strategy, Structure, Systems, Style, Staff and Skills. The Strategy is the only dimension that takes into consideration the external environment like competition and customers although it could be argued that at least the Structure dimension should reflect the external environment as well. The other dimensions, such as Structure, Systems, Style, Staff and Skills, focus on the internal organization of the company and especially how
the various units (divisions, departments) are structured and which systems and processes they adopt. The model is based on an iterative process aimed at identifying whether and to what extents the various aspects of an organization need to be aligned in order to improve performance over time.

The 7-S model (see Figure 2) can be used in a wide variety of situations where an alignment perspective is useful, for example, to improve the performance of a company, to examine the likely effects of future changes within a company, to align departments and processes during a merger or acquisition, to determine how best to implement a proposed strategy.

![Figure 2: The 7-S model](image)

Whatever the type of change – restructuring, new processes, organizational merger, new systems, change of leadership, and so on – the model can be used to understand how the organizational elements are interrelated, and so ensure that the wider impact of changes made in one area is taken into consideration. 7-S model is commonly used by managers to systematically analyse the organisational components and their interrelations.

**SWOT Analysis**

SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis is an important support tool for scenario analysis and decision-making, and is commonly used as a means to systematically analyse an organization’s internal and external environments.

By identifying Strengths, Weaknesses, Opportunities, and Threats, the organization can build strategies upon its strengths, eliminate its weaknesses, and exploit opportunities or use them to counter the threats (see D3.1 for a deepest description). The SWOT enable analysts to deeply analyse the impacts of an innovative solution: focusing on the internal factors that affect the success or the failure of this innovation, such as strengths and weaknesses elements that may affect its development, adoption and utilization. At the same time external factors that may affect the organization’s future (such as Opportunities and Threats) are identified by an external environment.
Comprehensive environmental analysis is important in recognition of the variety of internal and external forces with which an organization is confronted. These forces may comprise potential stimulants, and they may consist of potential limitations regarding the performance and/or the objectives that the organization wishes to achieve [9]. The obtained information can be systematically represented in a matrix and the different combinations of the four factors from the matrix aid in find strategies.

The most common elements analysts take into consideration as internal factors are:

- Competences and skill workforce
- Technological superiority or lacks
- Strong quality and development process
- Channels of communication and product/service distribution
- Low/high debt-equity ratio

Examples of external factors which may constitute opportunities or threats are:

- Changes in customer behaviours and tasting
- New players and new growing market
- New law and regulation
- Taxes
- International relationship among countries (such as the case of Brexit for instance)

Scenario planning

The scenario planning is a technique for making decisions in the face of uncertainty. The idea of scenario planning is to consider possible futures that include many of the important uncertainties in the system/organization. A scenario describes a possible situation, but the term has been used in a variety of ways [10] and the most common use of scenario refers to the expected continuation of the current situation. The idea behind the scenario planning is to establish future planning which can minimize surprises and broaden the span of managers' thinking about different possibilities [11], [12]. The result is a set of scenarios, which describe possible developments from the present to a given future time [13] and it may include also the sequence of interacting events needed to reach the scenario. Scenario planning uses not only facts on environmental and technological trends but also the perceptions held by decision-makers [14]. The characteristics of a good scenario are [15]:

- Relevance to top-management executives
- Internal consistency within each scenario
- Focus on different futures rather than variations on one theme
- A future scenario which will last for some length of time

The process of scenario planning can be conceptualized in four stages [16]:

1. Defining scope: by listing all the variables which should be taken into consideration. These include time frame, an industry or a market segment and stakeholders.
2. Database construction: The database is filled with predetermined elements, uncertainties, and trends [17].
3. Scenario building: exploration and extrapolation of predetermined elements [14], and different scenarios can come from the same set of predetermined elements.
4. Choosing strategic options: in-depth analysis of the scenarios and thinking through the organization's best actions and responses in each possible future.

The scenario planning can be defined by the whole prototype, which take into consideration all the views on a specific problem and is able to determine the connection among the views. This is also very coherent with the Balanced Score Card model described in D3.1.

2.1.2 Semi-automated and automated reasoning with BSc and SWOT

Since economic modelling involves representing complex systems, human-centric reasoning seems the intuitive answer. Human-centric reasoning includes both quantitative and qualitative reasoning: the first involves numbers, measurements and mathematical calculations, while the second involves the inspection of attributes and/or criteria that cannot be numerically defined, but depend on human senses such as colour, shape, or smell to mention a few. In organization studies, these qualitative elements are, among others, the organizational culture, workers’ motivations, human behaviours, vision, mission, etc. [18][19]. But, the issue with representing complex systems such as the ATM is that often there is a lack of precise quantitative data, and as such qualitative reasoning seems the intuitive answer. We see some of its applications in the following.

Qualitative Reasoning (QR) and Balanced Scorecard (BSc) were put together by Huegens and Zelewski [20] to capture the future behaviour of a company. The combined technique supports knowing the effects of taken actions onto the objectives in each perspective of the BSc (through the use of simulation) to get an overview of future development of the company. For this, the authors construct a strategy map as a constraint model that includes causal links between objectives of the BSc perspectives, and run qualitative simulation over it to determine which behaviour is possible from a given initial state when a new action is launched. The formal aspects are related mainly to the connection among perspectives, which can be measured via key performance indicators (KPIs). These provide a quantitative measure which enable decision makers to reason on the best local choice. Since it will be difficult to model the whole environment and define the Pareto efficient solution, the simulation will focus on a specific number of variables and changes, defining the so-called local best choice.

Qualitative and human centric reasoning are used in the area of economics and business administration to analyse dynamics and causality a Keynesian model [21]. The authors investigate causality explicitly, representing it in economic model to determine ordering (resulting structure of the economic system). They then propagate constraints over the model in order to simulate all the possible global qualitative states (behaviours of the system) that satisfy predetermined criteria. Their algorithm is implemented in LPA Prolog, and was tested not only with a Keynesian model but also with a dynamic sector macro model where the goods, money and labour markets interact. One of the limits of this model is that in an ATM system the scenario is very complex, and the most complete and realistic medialization should be provided. Moreover, focusing on macro components pushes the
decision maker to find a macro-equilibrium, which may be in contrast with the human behaviours of the actors involved. In this case a more micro-economic and behavioural analysis is required.

In business administration, Hinkannen et al. [22] propose a general framework for analysing qualitative reasoning systems using representations of qualitative differential equations and qualitative difference equations. Based on set theory, their QR framework describes fundamental concepts such as qualitative models and solutions, and relates them to the quantitative analogues of its underlying quantitative reference system (from which they derive qualitative descriptions). They analyse accounting systems using rules constrained reasoning, and further extend their theoretical framework to include an approach to qualitative optimization.

Feedlers and Daniels [23] present a formal framework for explanation and diagnosis of a company performance with both qualitative and quantitative information. The framework is model-based, and it is applied as part of a financial diagnosis systems to be used by investment analysts. In line with this approach is the XBRL [24] (eXtensible Business Reporting Language) is a freely available and global standard for exchanging business information. XBRL allows the expression of semantic meaning commonly required in business reporting, statements and annual reports. The language is XML-based and uses the XML syntax.

Yuan and Chiu [25] develop a case-based reasoning (CBR) system that assigns suitable weights to each level of balanced scorecard in order to support performance evaluation. Here, the intended use of BSC is to transform an organization’s mission and strategy into a balanced set of integrated performance measures. CBR is used to retrieve cases from historical operation records, and when faced with a new problem it searches from similar cases and reuses their solutions for the problem at hand. As such, this work proposes a three-level feature weights design to enhance the inference performance of case-based reasoning. The work makes use of a genetic algorithm (GA) to facilitate weighting all of the levels in balanced scorecard and to determine the most appropriate three-level feature weights. GA-CBR aims at defining appropriate weighting values per feature as this is crucial for an effective case retrieval.

As far as reasoning techniques concerning SWOT analysis are concerned, Ramooshjan et al. [26] propose a quantitative basis to analytically determine the ranking of the factors in SWOT analysis via Evidential Reasoning approach. This paper applies SWOT and Richard Rumelt’s criteria to evaluate strategies in Evidential Reasoning approach. The latter has been developed to support the multiple attribute decision analysis under conditions of uncertainty. It is based on Dempster’s rule for evidence combination and uses belief functions for dealing with probabilistic uncertainty and ignorance.

Menga et al. [27] propose a new approach to ranking the strategy alternatives via the SWOT analysis by making use of the Axiomatic Fuzzy Set (AFS) theory to find the best description of the alternatives and use the Evidential Reasoning (ER) approach in the light of AFS to rank the alternatives. This hybrid method (involving AFS theory and the ER approach) avoids having ‘values in between’ of the Expected Utilities, and obtains degrees of belief are directly from the database via AFS theory.

From the validation model presented in D5.1 it emerges that decision makers utilize models that are mainly based on ad hoc KPIs which focus on multiple attribute analysis. These can be unveiled and
adopted in the before-mentioned Balanced Scorecard model, business process reengineering practices or financial statements.

### 2.1.3 Conceptual modelling approaches to economic and organizational reasoning

The conceptual modelling community has dealt with the problem of representing business objects and processes for over a decade now [28]–[30]. These efforts have also resulted in standards, such as the business process modelling notation (BPMN\(^1\)). However, these works pay little attention to adequately capturing business objectives.

A first step towards addressing this issue is creation of the business motivation model (BMM\(^2\)), which provides a schema and structure for developing, communicating and managing business plans in an organized manner. BMM proposes an extensive vocabulary for modelling business objectives. Specifically, it: identifies factors that motivate the establishing of business plans; identifies and defines the elements of business plans; and indicates how all these factors and elements inter-relate. BMM should be used in combination with other techniques, such as BSC, to support decision-making.

Going back to conceptual modelling approaches, goal models have been proved particularly advantageous in conceptualizing strategic business problems and capturing viable alternatives in support of formal strategic decision-making [31].

The Business Intelligence Model (BIM) [32] is a goal modelling language expressly tailored for strategic modelling. BIM captures concepts found in strategy maps and BSc, such as strategic goals, performance measures, etc. It also supports representing situations, which are fundamental to supporting SWOT analysis, as well as indicators to measure performance. Most importantly, BIM supports special forms of reasoning, extending the work in [33] to take into account situations and indicators and helping analysts in choosing amongst alternative strategies. The resulting hybrid reasoning procedure combines indicator and goal modelling approaches, to allow an organization to answer strategic or monitoring questions, such as "Which strategy is better to achieve these goals?", "Which option is better for maintaining revenue growth and reducing risks?", etc. Answering such questions requires evaluating alternative strategies, assessing an operational strategy, inferring values for composite indicators, and so on.

Mate´ et al. [34] extend current strategic goal model analysis techniques [32][35][36] to support SWOT-based stress testing, which consists of evaluating a business strategy with respect to a series of scenarios. This work is motivated by the need to assess the adequacy of current strategies in light of an ever-changing and uncertain underlying environment, in which it is important to continuously maintain an updated view of the operating context. This proposal intends to improve decision-making by (i) supporting continuous scenario analysis based on current and future context and, (ii)

---

\(^1\) [http://www.omg.org/spec/BPMN/1.2/](http://www.omg.org/spec/BPMN/1.2/)

\(^2\) [http://www.omg.org/spec/BMM/1.0/](http://www.omg.org/spec/BMM/1.0/)
identifying and comparing strategic alternatives and courses of action that would lead to better results. This allows decision makers to identify what scenarios potentially lead to success or failure and what strategic alternatives are more tolerant to external factors.

2.2 Security reasoning techniques

As advocated in D3.1, in order to deal with security in the ATM domain, we need to follow a socio-technical approach, with the most prominent ones being based on goal-orientation. Therefore, here we provide a review of reasoning techniques for sociotechnical security approaches, overlooking system-oriented ones such as [37]–[40]. In addition, we consider security standards regulating security procedures and controls of an organization. They are crucial to ensuring the latter complies with international regulations.

2.2.1 Socio-technical security reasoning techniques

De Landtsheer and Van Lamsweerde [41] model confidentiality claims in terms of specification patterns, representing properties that unauthorised agents should not know. Their reasoning identifies violations of confidentiality claims in terms of counterexample scenarios present in requirements models. Diagnosis algorithms are used to generate the unauthorised agents reasoning to infer knowledge that is claimed to be confidential.

SI* [42], [43] proposes automated reasoning to check security properties of a model, and studies the interplay between execution and permission of trust and delegation relationships. Inconsistencies are identified as a result of considering two different levels, social and individual, based on the role-based access control model. They specify entitlements, objectives and responsibilities to roles, and then assign agents to roles. The social level represents for instance the policies that rule the organisation, while the individual level represents the concrete instance of the organisation. Therefore, inconsistencies are identified in case the concrete instance violates the specified security requirements.

Elahi’s work [44] extends the i* framework by supporting security trade-off analysis. The authors propose a conceptual modelling technique to reach a good enough security level in a multi-actor setting. This technique offers the possibility to assess the impact of assessing security mechanisms on actors’ goals and threats. Vulnerabilities refer to the deficiencies in the structure of goals and activities of intentional agents.

Paja et al. [45] propose the STS approach for modelling and reasoning about security requirements. In STS, security requirements are specified, via the STS-ml requirements modelling language [46], as contracts that constrain the interactions among the actors in the socio-technical system. In STS, a requester actor requires a requestee actor to comply with a security need (e.g., integrity, confidentiality, availability, authenticity, etc. in line with ISO27002 [47]). Each participant can express its own requirements, and the STS-ml model represents the business policies of the participants as well as their security requirements over information and goals. The requirements models of STS-ml have a formal semantics which enables automated reasoning for detecting possible conflicts among
security requirements as well as conflicts between security requirements and actors’ business policies. Automated reasoning is supported by STS-Tool\(^3\).

This work is extended by Meland et al. [48] to support threat modelling and analysis. In this approach, threats provide a foundation and justification for the security requirements that originate from goals. Tool support enables the reuse of threats facilitated by a threat repository, as well as allows for automated analysis techniques that not only propagate threats in the combined models, but also aid the analyst by visualizing the impact of threats on different elements of an STS-ml model such as actors, goals, and documents. This enables creating better requirements in the first place and thereby better solution designs and implementations.

Salnitri et al. [49][50] extend STS to further analyse and verify security requirements not only at the social and organizational level, but also at the business process level. They propose SecBPMN2, a modelling language for modelling business processes with security concepts, and procedural security policies. SecBPMN2 is composed of two parts, SecBPMN2-ml for modelling business processes with security aspects, and SecBPMN2-Q for modelling procedural security policies. Automated reasoning techniques permit to specify SecBPMN2 diagrams and verify them against security policies. Moreover, this work permits to automatically generate part of the implementation code in an artefact-oriented script language called River Definition Language [51]. All these activities are supported by a plug-in for STS-Tool [52].

### 2.2.2 International security standards

There are numerous international and industry standards on information security management [47], [53], [54]. They provide industry best practices addressing the organizational aspects of IT security and are used to drive the design and implementation of organizational and system processes.

A very well-known and wide spread standard is the ISO27002 standard [47]. It provides best practice recommendations on information security management. As such, it defines information security as the “preservation of confidentiality, integrity and availability of information; in addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved”. This standard provides a model for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving an information security management system. Information security is a main pillar in ISO27002, acknowledging the importance of information as an asset that needs to be suitably protected, like other important business assets. ISO27002 states that an appropriate protection of information requires its protection in any form (information can be printed or written on paper, transmitted via email, shown on films, or spoken in conversation) or means by which it is shared or kept (paper, email, film, or voice). The standard provides a set of (11) security control objectives—considered a good starting point, as they apply to most organizations or environments—to develop specific guidance to implement information security systems. These include risk assessment and treatment, security policy, organizing information security, asset management,

\(^3\) [http://www.sts-tool.eu/](http://www.sts-tool.eu/)
communications and operations management, access control, and information security incident management among others. The standard describes each considered control objective, and provides high-level guidelines for their implementation. However, it does not include more fine-grained details on how to implement these control objectives.

System Security Engineering Capability Maturity Model (ISO/IEC 21827) [55] was proposed as a framework for the evaluation of a security engineering process. ISO/IEC 21827 is aimed at facilitating the implementation of a good process for any set of security practices. Thus, the assurance roadmap provided by ISO/IEC 21827 helps identify Security Goals, Assess Security Posture, and Support Security Life Cycle. The standard assigns a level of maturity to a security system engineering process best on measurements performed on best practices, on the basis of which it suggests improvements, iterations, etc.

Common Criteria (CC) [53] is a computer security standard (ISO/IEC 15408), which provides a guide for the development, evaluation and/or procurement of IT products with security functionality. The CC addresses protection of assets from unauthorised disclosure, modification, or loss of use. The security categories relating to these three types of failure are commonly the CIA triad security protections, i.e., confidentiality, integrity, and availability, respectively. Common Criteria allows: (i) consumers to specify their security requirement (provides a set of common requirements for security functionality and assurance measures), (ii) developers to implement and make claims about the security attributes of their products, and (iii) evaluators to establish if the said products meet the claims, checking against evaluation assurance levels. CC is usually supplemented by other standards, such as ISO 27002, for it does not guarantee the security of products, instead it is meant to be used as the basis for evaluation of security properties of IT products. CC ensures that the process of specification, implementation, and evaluation of a product has been conducted in a rigorous and standard way.

NIST-IR 7298 [56] provides a glossary of information security terms. It defines Adequate Security as: “Security commensurate with the risk and the magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information. This includes assuring that information systems operate effectively and provide appropriate confidentiality, integrity, and availability, through the use of cost-effective management, personnel, operational, and technical controls.” Although not explicitly part of its information security definition, NIST-IR considers authentication and accountability two important security goals for information security. Given its nature as a glossary, NIST-IR is not meant to provide guidance to implementing the defined security requirements.

COBIT [57] from ISACA, defines information security as: “Ensures that within the enterprise, information is protected against disclosure to unauthorised users (confidentiality), improper modification (integrity) and non-access when required (availability).” However, COBIT is not meant to serve as a standard tackling information security alone, rather it aims to provide best practices for enterprises for information technology governance and management. It provides a comparison with other available standards, such as ISO27002 [47] and NIST-IR [56], while providing directions to executive agencies on how to apply and adapt its good practices to specific contexts.
As far as our knowledge goes, there are no recognized standards for security and aviation. One notable exception is BS EN 16495:2014, though it is heavily based on ISO27002. The Aeronautical Telecommunications Network (ATN) was first tested for security by EUROCONTROL in 1995. The identified threats, including masquerade of a controller, that of a pilot, modification of uplink/downlink messages, and denial of service, resulted in the development of the ATN Security Extensions incorporated in ATN SARPs edition 3. Extensions require providing highly efficient mechanisms for authentication of the sender, protecting integrity and against misdelivery. Nevertheless, the shift from the inherently insecure ATC voice based system to datalink communication as part of Link2000+ programme has resulted in a general decrease of ATC vulnerabilities, especially concerning denial of services. The security risk assessment [58] developed as part of datalink is based on SecRAM [59] and is in line with ISO/IEC 27005 [60]. There are currently more obstacles for attackers to launch successful masquerades. As a result, the thorough implementation of ATC security extensions is delayed since the cost of deployment is not justified by the calculated risks.

2.3 Safety reasoning techniques

We present the state of the art on reasoning techniques specifically for the safety perspective. This includes a subset of the safety techniques presented in D3.1, which offer reasoning support. Safety reasoning regulated by standards and regulations do not prevent the use of formal, model-based, automated or informal, people-centric safety reasoning techniques. In addition to safety reasoning regulated by European or International standards or regulations, there are safety reasoning techniques that to a lesser degree have been used in practice, which are results from research projects and which have not yet been included in standards. Reasoning techniques from both areas are discussed in in this chapter.

Within several other application sectors, such as the railway, offshore and process industry, there exist standards and regulations that regulates the safety reasoning to a certain extent (ref. the CENELEC standards EN 50126 [61], EN 50128 [61], EN 50129, Common Safety Method (EU) 402/2013 [62], IEC 61508 series [63] and a number of other application sector specific standards). Several of these standards require that there exist processes for quality/safety management and further that sufficient argumentation that the technical safety of the developed system is achieved exist. All these three parts (quality management, safety management and technical safety) together with a conclusion with respect to the safety level achieved may be regarded as part of the safety reasoning. In the following description of safety reasoning we actually pay particular attention to the standards for railway. The railway standards are among those which have paid most attention to complex contexts of changes which may impact several stakeholder perspectives. For instance, the CENELEC standards cover how to deal with security issues within a safety-critical context, whereas IEC 61508 does not. Due to the fact that demonstration of compliance with safety standards are considered to
be a mature and established approach within the railway sector, these standards are selected as base for further investigations regarding safety reasoning.

There are also standards for ATM, including ED-12/DO-178\textsuperscript{5} which covers software assurance for airborne systems and equipment certification, and ED-109/DO-278\textsuperscript{6} is applied for non-airborne communication, navigation, surveillance and ATM systems. COMMISSION REGULATION (EC) No 482/2008 on establishing a software safety assurance system for air navigation service providers, says that traceability is addressed in respect of all software safety requirements, but not much in relation to reasoning techniques. Yet much knowledge from the aforementioned field is directly applicable to the ATM domain, core concerns are shared between domains, and in the context of reasoning we do not encounter domain specific matters at this level anyway.

The safety reasoning techniques presented in the following sections are based on the COMMISSION IMPLEMENTING REGULATION (EU) No 402/2013 of 30 April 2013 on the common safety method for risk evaluation and assessment and repealing Regulation (EC) No 352/2009, and CENELEC standards EN 50126, EN 50128, EN 50129. EN 50129 which specifically requires documentation of the safety evidence within a safety case has been around since 2003.

2.3.1 Safety reasoning regulated by CENELEC standards

As already mentioned, The CENELEC standards require that there exist processes for quality/safety management and further that sufficient argumentation that the technical safety of the developed system is achieved exists. These three parts together with additional parts (definition of system, related safety cases and conclusions) are included in a document denoted a safety case. A subset of the safety techniques presented in D3.1 represent techniques that are used as part of the executing the safety management processes and techniques for arguing that the technical safety is achieved.

According to these three CENELEC standards, these three parts together with the together with the additional parts included in a safety case are evaluated by an independent third party, denoted an independent safety assessor. The independent safety assessor concludes with respect to compliance with these standards and with respect to the achieved safety level.

The above mentioned standards and regulations are further equally applicable in the context of designing a complete new system as for implementing changes within existing systems (change management); as an example of this EN 50129 states that: "This standard applies to the specification, design, construction, installation, acceptance, operation, maintenance and modification/extension phases of complete signalling systems, and also to individual sub-systems and equipment within the complete system."

\textsuperscript{5} ED-12 and DO-178 the same standards, published under different names in the US and Europe http://standards.globalspec.com/standards/detail?docid=1495165&familyid=LVXUTCAAAAAA

\textsuperscript{6} The same applies to ED-109 and DO-278 http://standards.globalspec.com/std/1517993/eurocae-ed-109
In the following a short description will be provided of the three different parts (quality management, safety management and technical safety) that represent major parts of the safety reasoning. As part of the descriptions provided it will be made clear whether the safety reasoning can be supported by model-based, automated approach or manual, people-centric approach. It is worth pointing out that all three parts may be supported by using tools.

A short description will also be provided regarding a few of the possible safety techniques, ref. D3.1, which can be used as part of the safety reasoning and the role of an independent safety assessor.

**Quality Management**

According to the standard EN 50129, a first condition for safety acceptance shall be that the quality of the system, sub-system or equipment has been, and shall continue to be, controlled by an effective quality management system throughout its life-cycle. The standard further state that the purpose of the quality management system is to minimise the incident of human errors at each state in life-cycle and therefore reduce the risk of systematic faults in the system, sub-system or equipment. The standard lists aspects that should be controlled by the quality management system. It’s worth noting that, the standard does not require how the mentioned aspects shall be controlled. It is evident from aspects listed that they to a certain degree must be controlled by manual people centric procedures, but in a similar manner it also evident that parts of the aspect can be controlled by using appropriate tools. The manufacturer of a system may therefore choose to cover those aspects by a combination of manual people centric procedures and by using tools.

**Safety Management**

According to the standard EN 50129, the second condition for safety acceptance of the system, subsystem or equipment has been, and shall continue to be, managed by means of an effective safety management process. According to EN 50129, “the purpose of this process is to further reduce the incidence of safety-related human errors throughout the life-cycle, and thus minimise the residual risk of safety-related systematic faults.” Aspects that the safety management process shall include are specified by the standard. The manufacturer of the system being developed or changed can choose to cover those aspects by manual people centric procedures and by using appropriate tools. With respect to Regulation 482/2008 on Software Safety Assurance in ATM [64], the objective is to ensure that Air Traffic Services (ATS) providers implement within the framework of their Safety Management Systems (SMS) a software safety assurance system to manage and reduce risks associated with the use of software in the European Air Traffic Management network systems (EATMN software) to a tolerable level. The Regulation lays down the requirements for the definition and implementation of a software safety assurance system by providers of ATS, air traffic flow management (ATFM) and air space management (ASM) for general air traffic, and providers of communication, navigation and surveillance (CNS) services.

A major part of the safety management is to identify the safety requirements. According to EN 50129, the safety requirements shall be identified by means of hazard identification and analysis, risk assessment and classification, allocation of safety integrity levels. The hazard identification and risk assessment, which is performed in order to create the basis for the safety requirements, can be performed by using a subset of the safety techniques described in D3.1. It shall be noted, however,
that a subset of the safety techniques described in D3.1 can also be used as part of the evidence that the technical safety is achieved; see immediately below.

**Technical Safety**

In EN 50129, the third condition for safety acceptance of system/sub-system/equipment is the technical evidence for the safety of the design. The standard requires that aspects such as **assurance of correct functional operation, effects of faults, operation with external influences, safety related application conditions** and **safety qualification tests** shall be described and handled in a sufficient manner. Further detailed requirements regarding the mentioned aspects are presented within EN 50129. Regulation 482/2008 [64] also identifies and adopts the mandatory provisions of the EUROCONTROL Safety Regulatory Requirement, ESARR 6 Software in ATM Functional Systems [65]. This Regulation applies to the new software and to any changes to the software of the systems for ATS, ASM, ATFM, and CNS. It does not apply to the software of airborne and space-based equipment.

As part of the evidence needed for technical safety, EN 50129 requires that software development shall be performed according to requirements of the standard EN 50128. Regarding software development, the standard EN 50128 acknowledges that tools are necessary to help developers in the different phases of software development. According to EN 50128 example of relevant tools may be: transformation or translation tools (design refinement tools, compilers, assemblers, linkers, loaders and code generation tools), verification and validation tools, diagnostic tools, infrastructure tools, configuration control tools, application data tools. The different tools may be assigned to one of three defined tools classes within EN 50128: tool classes T3, T2, T1. The standard therefore permits the use of tools, however, evidence shall be provided that tools are sufficiently qualified by presenting the evidence needed according to the selected tool class. The strictest tool class defined in EN 50128, is class T3. Examples of class T3 tools according to EN 50128 are source code compiler, data/algorithms compiler, a tool to change set-points during system operation, an optimising compiler where the relationship between the source code program and the generated object code is not obvious, a compiler that incorporates an executable run-time package into the executable code. If using class T3 tools as part of software development, evidence shall be presented that the output of the tool conforms to the specification of the output or failures in the output are detected. Further detailed requirements are included in the standard on what is needed in order to provide the evidence for this requirement. Examples of class T2 tools may include test harness generator, test coverage measurement tool, a static analysis tools. Examples of class T1 tools may include text editor, requirement or design support tools with no automatic code generation capabilities etc.

EN 50129 further presents a number of requirements related to the **effects of faults** aspect of technical safety. A subset of the safety techniques identified in D3.1 may be used in order to provide some of the evidence needed to demonstrate the **effects of faults**. A few of those techniques are discussed below.

To sum up regarding the evidence for technical safety, parts of the evidence may be provided by executing manual people centric procedures and parts of it by using tools. However, as described above the use of tool implies that sufficient evidence is provided for the tools regarding compliance with the requirements according to the appropriate tool class.
2.3.2 Safety techniques

In the following a subset of the safety analysis techniques reported in D3.1 are discussed and investigated with respect to reasoning support.

The safety techniques described below may be used within a subset of the stages of the eight stage safety assessment process presented in D3.1.

**Fault Tree Analysis:**

Fault tree analysis is defined as a highly recommended technique according to EN 50129 and be used as part of the argumentation for *effects of faults* aspect of technical safety and as such contributes to the safety reasoning. According to Rausand [66], the fault tree analysis is a logical diagram that illustrates the relationship between an unwanted event top event in a system and the cause of the event. A fault tree analysis can be both qualitative, quantitative or both. The analysis can provide answers to the following questions: which combinations of faults an events can lead to the top event, how probable is the occurrence of the top event, how often will the top event occur, which component and/or event will be the major contributor to event occurring or not?

Rausand [66] presents the advantages of the technique. A subset of these advantages is:

- It gives a clear picture of which combination of equipment failures and other events that can lead to the top event,
- It is the most used technique within risk analysis,
- Those performing the analysis are forced to understand the system thoroughly, and as such several weaknesses of the system developed can be discovered and corrected already while developing the fault tree.

According to Rausand, the disadvantage of the method is that it gives a static picture of the combinations of failure that can lead to the top event.

In order to provide quantitative calculations relevant experience data must be available.

A number of tools exists that can be used for creating fault trees. The analysis in the context of providing evidence for the *effects of faults* aspect of the technical safety is that it may be used to provide evidence for the rate of occurrence of the top event and as such contribute to the demonstration that the system/sub-system/equipment continues to meet its specified safety requirements in the event of random hardware faults.

**FMECA:**
A FMECA is mainly a qualitative analysis which is normally performed in the design phases of a technical system [66]. The analysis can have certain quantitative elements such as frequency of the failure modes together with determination of the seriousness of the fault effects. The analysis can also be performed in later phases of a system’s lifecycle. The purpose of the analysis is usually to identify elements or characteristics of the system that should be improved in order to comply with safety or reliability requirements. The analysis is performed on component level, however, as part of the analysis the effects on the remaining parts of the system shall also be evaluated.

According to EN 50129, FMECA can be used together with fault tree analysis as a means of providing the evidence of effects of faults aspect of the technical safety.

Other safety techniques:

EN 50129 describes further techniques, in addition to the two mentioned above, that are highly recommended in terms of providing evidence of the effects of faults aspect of the technical safety; these are: preliminary hazard analysis, Markov diagrams, hazop, interface analysis and common cause failure analysis.

Independent Safety Assessor:

EN 50129 states that an independent safety assessment of the system/sub-system/equipment and its safety case needs to be carried out before the application for safety approval can be considered. The safety case shall report the performed quality management, safety management and include the necessary evidence for the technical safety in addition to other aspects such as definition of system, related safety cases and conclusion. EN 50129 states that "the depth of the safety assessment, and the degree of independence with which it is carried out, are based on the results of the risk classification, as explained in EN 50126. Specific tests may be required by the safety assessor in order to increase confidence." The report produced by independent safety assessor shall explain the activities carried out by the assessor to "determine how the system/sub-system/equipment, (hardware and software) has been designed to meet its specified requirements, and possibly specify some additional conditions for the operation of the system/sub-system/equipment."

Even though EN 50129 requires that the safety evidence shall be evaluated by an independent safety assessor, the level of the details that the assessment is to be performed is not defined. This openness means that there is room for discussion and mutual adjustments, something we will describe further in section 3.2.

2.3.3 Safety reasoning regulated by COMMISSION IMPLEMENTING REGULATION (EU) No 402/2013

COMMISSION IMPLEMENTING REGULATION (EU) No 402/2013, abbreviated CSM hereinafter, is a European Regulation which provides a common safety method for risk evaluation and assessment. CSM provides the criteria for determining whether a change can be considered to be a significant change. If the change is considered to be significant, a risk management process compliant with CSM must be followed which also implies an assessment performed by an independent assessment body.
If the proposed change has an impact on safety, the proposer shall by expert judgement decide on the significance of the change based on the following criteria: failure consequence, novelty used in implementing the change, complexity of the change, monitoring, reversibility, additionally.

These criteria present the possibilities of a change having an impact of safety, but at same time being a non-significant change and therefore not requiring an assessment of an independent assessment body. In PACAS this would translate to judging whether to choose changes that require assessment or merely internal reviews, which in itself may have economic impacts.

### 2.3.4 Safety reasoning beyond standards

While safety-critical systems are most closely connected to safety standards, we also present some thought and ideas of various additions and comments on them. We describe in brief how optimal maintenance takes more than safety into consideration, an extension to safety cases to enhance understanding, and the use of formal methods.

Due to increased system complexity and rise of system criticality, development of model-based safety analysis methods has been a focus for many years. However, in real-world applications, minimization of hazard rates is not the only design goal. Within several domains, there are trade-offs between safety and other factors such as preventive maintenance effort, availability etc. These trade-offs are often made due to economic reasons, and effort has been made to study how to optimize safety and other factors using cost as the basis for comparison. An increasing number of models link for instance maintenance and safety risks. It is acknowledged that safety is a main concern for determining the right level and type of the maintenance activities, see for example Vatn et al. [67] and the references therein, Lyngby et al. [68] and Lyngby [69]. These present an approach for identifying the optimal maintenance schedule for the components of a production system. Safety, health and environment objectives, maintenance costs and costs of lost production are all taken into consideration. Lyngby et al. [68] presents different models for maintenance/renewal optimization, and Lyngby [69] describe a Markov state model for optimization of maintenance and renewal of infrastructure, using the parameters like cost of renewal, preventive maintenance, corrective maintenance and accidents in the optimization model.

Development of safety-critical systems requires adherence to various standards according to domain. While hardware system components can be measured with a quantified associated risk, this is more difficult within the software domain. The development process and final product still must adhere to varying levels of safety integrity (SIL) or similar. In todays' competitive and fast-changing world, tailored hardware components are gradually replaced with standardized hardware components, which again run tailored and increasingly complex software systems. The assumption is that the higher the SIL (0 through 4), the less prone the component is to critical failures. Weaver et al. questions this assumption, arguing that this process focusses on generating evidence while failing to consider the required objectives [70]. They argue that the evidence not necessarily quantitatively demonstrate that the SIL has been achieved.

Kelly and Weaver state that while safety cases are can be good to in a clear way argue that a system is acceptably safe given a particular context, they are often written poorly [71]. Safety cases are based on a lot of prose, and safety cases of low quality make it difficult to convey a shared
understanding of the arguments for safety. Likewise, logical fallacies in the arguments of a safety claim may undermine its safety claims [72]. Kelly and Weaver suggest using a Goal Structuring Notation (GSN) – a graphical argumentation notation. Goals, solutions, strategies, contexts and undeveloped goals are mapped in a model. According to the authors, use of GSN gains momentum, and they think one should consider applying it in other domains where assurance cases are required, such as security.

2.4 Multi-view multi-objective reasoning techniques

Since we are dealing with change through a multi-view multi-objective approach, we need to understand the implications that changes in one perspective have not only within the same view, but also over the rest of the views. A first step to finding a trade-off, is to identify those solutions that bear no conflicts, that is why we analyse reasoning techniques for conflict identification first to then continue with techniques for trade-off analysis.

2.4.1 Reasoning for conflict identification

A prominent body of work that offers support to reason for conflict identification is in the requirements engineering field, where in particular goal modelling techniques have emerged to represent stakeholders’ objectives/needs/requirements. Thus, we review the literature in this area.

SAT solving was applied to analyse the satisfaction or denial of goals in goal models [73]. Giorgini et al. define a range of satisfaction and denial evidence, from full to partial, resulting in four distinct predicates: full evidence of satisfaction (FS), partial evidence of satisfaction (PS), full evidence of denial (FD), and partial evidence of denial (PD). By introducing the positive and negative contribution relationships among goals, they can identify situations of contradictory contributions among goals. The new relationships are given both a qualitative semantics and a quantitative semantics, which are based on a probabilistic model. As such, the approach includes both qualitative and quantitative analysis techniques that determine evidence of goal satisfaction/denial by using label propagation algorithms, which are proven to be sound and complete. Conflicts are identified when both positive and negative evidence exists.

In [74], Giorgini et al. incorporate goal model analysis procedures introduced in [73] into the Tropos Framework, with the objective of making the goal analysis process concrete while coping with qualitative relationships and inconsistencies among goals, in order to suggest, explore and evaluate alternative solutions. Goal analysis includes forward reasoning, which is used to evaluate the impact of the adoption of different alternatives (leaf goals being fulfilled) with respect to the softgoals of the system to be. Additionally, backward reasoning is used to find the acceptable alternative (set of leaf goals) at the lowest costs, such that if achieved can guarantee the achievement of the desired root goals and softgoals. As far as the identification of conflicts is concerned, goal analysis identifies conflicts not only based on contradictory contributions to the same goal as in [73], but also possible due to the existence of diamonds or loops.

This approach inspired further research, such as the iterative and interactive identification of conflicts [75][76] by Horkoff and Yu for early requirements engineering. In [75], the authors propose
a qualitative interactive procedure that allows users to systematically evaluate and compare the alternative actions and solutions expressed in models asking “What if?” questions. The modeller can supplement the evaluation with domain knowledge, to guide model creation and domain exploration, which includes the evaluation of alternatives. In this way the procedure can benefit from human intervention to compensate for the incomplete nature of the models, while reasoning in early stages of analysis, before concrete quantitative information is known. The procedure propagates forwards by first deciding on an alternative, second propagating satisfaction/denial labels and placing results over softgoals (often receive multiple incoming labels), and finally resolving conflicting labels over softgoals either through automatic cases or human judgment when cases do not apply. In this way, the procedure does not itself decide on an alternative, but picks a case or human judgment which leads to selecting an alternative, while evaluating the effects of alternative choices in the model.

The work described in [76] on the other hand, proposes an interactive backward procedure from ends to means to answer questions “Is this possible?” “If so, how?” and “If not, why not?”. They encode i* models (used as an example modelling language) and target values into conjunctive normal form (CNF), and use a SAT solver to reason on this representation. The SAT solver is called iteratively on the CNF representation. After each iteration their approach takes user input to take a decision in case of conflicts or multiple sources of partial evidence, by using domain knowledge, to then re-encode CNF formula removing the axioms for backward or forward propagation, and adding new axioms representing human judgment. When the SAT solver finds the answer, it returns success, and human judgment is not needed. When the SAT solver cannot provide an answer, they display UNSAT and backtrack over the last round of human judgment to find an answer. Should no more human judgment be available to backtrack over, then the procedure returns failure, for no answer was ultimately found.

Fuxman et al. [77] translate i* models to Formal Tropos, which supplements i* concepts with first-order linear-time temporal logic. The use of formal methods, in particular model checking techniques, intends to allow a formal and mechanized analysis of early requirements specifications to help the requirements analyst identify errors and limitations of the specification, which would be otherwise impossible in an information setting. Formal assertions are used to represent a set of required and desired constraints over the system. The framework uses an intermediate language to link Formal Tropos and model checking. Early requirements are translated into this intermediate language and given in input to a symbolic model checker to verify contradictions in the requirements specification (consistency checking), as well as to validate formal properties (for actors, goals or dependencies). When identifying property violations, their analysis returns concrete conflicts’ scenario (counterexample) that describes the scenario violating the property, to help the user understand the problem.

KAOS [78] includes analysis techniques to identify and resolve inconsistencies that arise from the elicitation of requirements from multiple stakeholders with different viewpoints. Emphasis is put on formal analysis that identifies goal conflicts. The authors review various types of inconsistency classifying them based on the description of the requirements, identifying: process-level deviations, which refer to a state transition in the requirements engineering process resulting in an inconsistency between a process-level rule and a process state; instance-level deviations, which results in an inconsistency between a product level requirement and a specific state of the running.
system; terminology clashes, in which the requirements specification contains different syntactic names for the same concept; designation clashes, when a single syntactic name in the requirements specification designates different real-world concepts, and structure clashes, when a single real-world concept is given different structures in the requirements specification. Conflict is defined among several specifications of goals/requirements enhanced with domain knowledge, which seems to play an important role in the identification and anticipation of conflicts (logical inconsistent assertions). Importantly, resolution techniques are discussed, such as finding alternative goal refinements, weakening goals, and using of divergence (a boundary conditions that makes assertions logically inconsistent) resolution heuristics. The proposed classification and resolution techniques are shown to be useful in managing inconsistencies in a real-world project; however most of the classifications and resolution strategies are specific to KAOS concepts, such as boundary conditions.

2.4.2 Finding trade-offs among multiple objectives for decision-making

Typically, finding a trade-off among various objectives is a challenging optimization task, especially due to the heterogeneous nature of objectives and the interdependencies that exist between them, that is, the impacts they have on one another.

Nguyen et al. [79] extend goal models to express constraints via numerical attributes as well as preferences between goals and refinements, and with the use of automated reasoning technologies, notably Satisfiability and Optimization Modulo Theories (SMT/OMT), they support not only satisfaction analysis but also multi-objective optimization analysis. The framework uses OptiMathSat [80] as the external SMT/OMT solver to find solutions.

Aydemir et al. adopt the CGM approach of Nguyen et al. to analyse risk in goal models [81]. Their multi-objective goal-oriented risk analysis framework includes inter-dependencies among treatments and risks in terms of likelihood and it generates optimal solutions by taking into account multiple objectives such as goal rewards, the costs of treatments, or risk factors.

Algorithmic techniques for decision-making are designed to automatically select the best alternative between two or more available options based on multiple criteria. They provide a formal framework to choose among multiple alternatives.

The largest body of work is centred around MCDM (Multi-Criteria Decision-making) [82] [83] [84], where the Analytical Hierarchy Process is included [85], and there are multiple surveys comparing the different techniques [86]. MCDM techniques have been applied to quantitative [82], fuzzy [84] or qualitative [86] alternative selection in various domains, including vendor selection [83] or supplier assessment [85]. The basis for MCDM techniques consists in listing out the alternatives that can be selected and eliciting how each alternative relates to criteria at hand. Then, using the relative importance of the criteria, alternatives are ranked and the selection of the best alternative is made.

In the case of a finite number of criteria and undefined (infinite) number of the feasible alternatives (the ones meeting the requirements), multiple criteria optimization and multiple attribute decision analysis (MADA) are very effective [87]. A multiple attribute decision analysis (MADA) problem can be generally modelled using a decision matrix [86], as shown in Table 1. Each row belongs to a
criterion and each column describes the performance of an alternative. The score $A_{ij}$ describes the performance of alternative $A_j$ against criterion $C_i$. For the sake of simplicity, we assume that a higher score value means a better performance since any goal of minimization can be easily transformed into a goal of maximization.

<table>
<thead>
<tr>
<th>Table 1: Multi-attribute decision-making</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$W_1$</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$W_m$</td>
</tr>
</tbody>
</table>

As shown, weights $W_1, \ldots, W_m$ are assigned to the criteria. Weight $W_i$ reflects the relative importance of criteria $C_i$ to the decision, and is assumed to be positive. The weights of the criteria are usually determined on subjective basis. Usually, higher ranking values mean better performances of the alternative, so the alternative with the highest ranking value is the local best of the alternatives.

The main drawback of algorithmic techniques is that they only explain the impact of the selection on the criteria used as an input. The effects of the decision for the different units within the firm or outside the firm remain unknown, and thus can be difficult to accurately assess. Additionally, MCDM techniques stem from the idea that all criteria and relationships with respect to the alternatives at hand are well-known and established. In cases where uncertainty is present, such as when new technologies appear, these techniques cannot explain how the alternative that was chosen under a different set of assumptions will perform.

To solve the second drawback, some approaches, like Letier's to select the most adequate architecture [88] include uncertainty and information value into the analysis. However, due to the nature of the algorithmic approaches, the first drawback still remains uncovered.
3 Identification of the needs

Similarly to what was described in D3.1 Section 3, in order to obtain a valid view on the requirements of ATM stakeholders with respect to reasoning techniques, PACAS uses an iterative approach to elicit needs from Advisory Board (AB) members representing the stakeholder experts, as well as researchers in the project who have practical experience with working in the domain. This chapter reports on preliminary results from early interviews with AB members (Sections 3.1 and 3.2) as well as some experiences from reasoning in the ATM domain in practice (Section 3.3). At current state, the interview findings constitute a preliminary report as the results are reported from a high level point of view as the analysis of the AB workshop is in progress. The depth of our understanding and analysis will however increase along with each iteration of the project proposing concepts and design hypotheses to explore with stakeholders.

3.1 Interview method

We report a brief version of the methodology used for the identification of the needs of the stakeholders with respect to the reasoning techniques for decision-making. Part of this section refers to the work already described in D3.1 Section 3 for details on the identification of the needs and in D5.1 for further details on background.

It is important to remind here that PACAS is based on the end-to-end participation of AB members representing the ATM stakeholders. In particular, each AB member will help building and integrating the models providing background, experience and know-how from his/her own sub-domain perspective (view). The first phase of the project focuses on the analysis of the gap in large-scale systems design and reasoning methodologies, ending with the relative validation activity of elicitation of the needs and requirements of the stakeholders (please refer to D5.1 for the entire validation strategy and plan).

The research technique used for data collection at this stage of the project consists of semi-structured interviews (already represented in D3.1 and 5.1 Section 3) and brainstorming activities with the AB members (made during the first Validation Workshop in Rome, 4th June 2016). Section 3.2 shows the first insights from this activity.

3.2 Results from interviews and Validation Workshop #1

ATM systems are very complex, interrelated and need to be continuously updated and modified. The implementation of any change and therefore also the adoption of new procedures or systems are
closely related to the interdependencies of ATM systems even at international level. These changes can be addressed by political decisions (new regulations, new international needs, etc.) or by technical requirements (innovative technologies, etc.). What has emerged during the interviews and the workshop in Rome is the fact that change management in ATM means re-arranging both directly and indirectly involved aspects at any level: systems, procedures, services, quality, etc. As a consequence, any change is always complex: it may initially be tested in a specific geographical area, or a specific operational function, in order to evaluate the impact in a specific setting, but after a while it might affect the whole system.

Hence, any change should be deeply analysed and multiple view reasoning models help in providing useful insights on various perspectives, and interrelation among the different views.

The most relevant issues emerged by the interviews are related to change management and innovation of various perspectives which can be summarised as follow: (i) safety and security management; (ii) organizational and costs management; (iii) efficiency and holistic analysis of complex systems.

For safety and security management, the main impression is that workshop participants perceived the reasoning support of the PACAS platform as a key selling point for our project. Considering that there is currently little support available for reasoning activities across stakeholder views, they acknowledged that there is a need for computer-based support in making the best decisions as such.

From organizational and costs management we should consider that organizational investments in developing procedure, improving workforce competences, and cost reduction are the common arguments discussed during the interviews and the workshop. Most of the procedural changes proposed in the past, are aimed at reducing costs and at making the system more efficient. In order to calculate the effectiveness of the new solution, KPIs are identified and used to validate the ideas.

Referring to the holistic analysis of complex systems, it emerged that decision makers utilize very naïve models, mainly based on ad hoc KPIs that have a very broad focus on multiple attribute analysis. In our opinion, modelling balance scorecard, using SWOT analysis, and evaluating the statement via XBRL may provide useful insights and views to deal with more efficient solutions.

To conclude we believe that a Multi Attribute Decision-Making model can be used to define the solution which, taking into consideration the various views in PACAS, can find a more effective solution (or Pareto efficient).

3.3 Needs from experiences on reasoning in the ATM domain

3.3.1 Identifying similarity and relatedness

One of the key problems with multi-view modelling (and with enterprise architectures in general) is that various models or views may refer to the very same concepts but without having an explicit, direct link between the models. For example, a safety model may represent an event such as strong wind while a performance model could refer to the effect of airstream on the speed of an airplane. In
order to keep the models aligned and consistent, it is important to identify these somewhat hidden relationships.

The key to do so is to employ natural language processing (NLP) techniques that can be executed to find entities having similar or related labels. Some types of relationships that can be identified are synonyms and hyponyms (concepts connected by the generalization relationship, such as woman and human). The most promising set of techniques to identify these links are based on semantic distance, which is a measure of how close or distant two terms are in their meaning [6].

We can distinguish two types of semantic distance: semantic similarity and semantic relatedness. The former measures how much two concepts are alike, based on information contained in an is-a hierarchy [4]. For example, “Apples” and “Bananas” share the common ancestor “Fruit”. Or, considering the aerospace field, and taking an example from the ATM Information Reference Model (AIRM), TakeOffWeight and MaximumTakeOffWeight share the common ancestor AircraftWeight.

Unlike semantic similarity, semantic related concepts do not have to share properties. This technique looks beyond similarity and looks for semantic relations between concepts [4]. For example, “Door” and “Knob” are semantically related as one is part of another (although the terms are not synonyms). On the other hand, “House” and “Bridge” are not semantically related one another. Taking again an ATM example from AIRM, AircraftAvionics is semantically related to Aircraft.

Both semantic similarity and semantic relatedness of word pairs are typically expressed as a number—usually in the [0,1] range—that captures the distance between the two words, with 0 being no relatedness and 1 being full relatedness. For any given word, this technique can be used to identify a list of similar words or to calculate its semantic similarity score with a collection of words. Repeating this process for all words in a collection, one obtains a matrix that defines the pairwise similarity between all those concepts. This matrix can be useful to relate the concepts from two ATM models/views, for example.

Different approaches exist to determine a value for these values. One of the earlier techniques was proposed by Hirst and St-Onge [3], whose technique for measuring semantic relatedness is based on an idea that two concepts are semantically close if the path between them in a given ontology is not too long and does not change direction too often. This measure calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies (a synset is a group of data elements that are considered synonyms), along with the depth of the Least Common Subsumer (the most specific concept that is common ancestor of two concepts). For the purposes of PACAS, one can consider replacing or complementing the general-purpose WordNet ontology with the AIRM.

Among the many existing approaches to calculating semantic similarity [2], some key advances in NLP seem to be especially interesting in terms of performance: skip-gram by Google [1] and GloVe by Stanford [5]. Both algorithms are based on the same principle: they parse enormous quantities of unannotated text to generate word embeddings without requiring supervision. A word embedding maps some attributes of a word to a vector of real numbers that can then be used for a variety of tasks, including the identification of relatedness. Intuitively, the idea is that two terms will be related when their vectors have several attributes in common.
These techniques are the most accurate state-of-the-art and provide significant improvements on traditional approaches. The positive news about these works is that they are quickly transiting from an academic discovery to an industrially applicable innovation, especially thanks to the recent development of easy-to-use tools that can be used to determine semantic relatedness, such as the spaCy tool\(^7\) that implements semantic relatedness based on word embeddings in an efficient manner. We are planning to experiment the extent to which these techniques, alongside the information provided from the ATM domain such as that in AIRM, can contribute to identifying non-trivial links between multiple views.

### 3.3.2 Past experiences

In many cases when safety reasoning is based on compliance with safety standards, an assessment performed by an independent safety assessor is required. The level of detail required for the assessments to be performed is in some cases not strictly defined. The assessment results may be based on evaluation of processes followed, evaluation of the competence of involved personnel, evaluation of technical documentation, evaluation of test results, evaluation of tools being used and their produced results. The usage of tools as part of design/development but also a part of verification/validation may help the independent safety assessor to gain trust in the development process and in the developed system provided the tools themselves have been assessed and considered to be qualified. In such cases, the tools themselves may be considered to be less likely to produce erroneous results compared to manual people centric operations. It may also save time and effort. SINTEF has experiences from working with IEC 61508 where companies have made use of Jira\(^8\) included add-ons as a key component in software development, both to control work and to automatically create and store relevant documentation related to this work. Considering that the tools have been adequately qualified and assessed by the independent safety assessor, they may ease the checks provided by the independent safety assessors. It is important to note however, that there are number of safety standards which may have different kinds of requirements/restrictions on the use of tools and development methodologies. Some standards are less method-and tool-agnostic than others.

Concerning safety reasoning, evaluation of the significance of the changes of a system may be based on the evaluation of the six criteria provided in CSM [62]. These criteria are failure consequence, novelty used in implementing the change, complexity of the change, monitoring, reversibility, and additionality. For instance, if a suggested change in ATM is a novel approach within aviation, it needs closer inspection. A safety reasoning tool may be helpful in that context for making the evaluation of the six criteria and concluding on the significance of the change. If the change is concluded to be non-significant, an assessment performed by an independent assessment body is not required. Being able to quantify these criteria may enable other parts of PACAS to reason about safety changes.

\(^7\) [https://spacy.io/](https://spacy.io/)

\(^8\) Jira Issue and Project Tracking Software: [https://www.atlassian.com/software/jira](https://www.atlassian.com/software/jira)
Traditionally, systems are verified and validated through the use of testing. Test cases are created to the largest possible extent to mimic complete system behavior and to cover the system requirements. Testing is performed during the different phases of the system development life cycle, from the testing of isolated functions and components in the lab to full scale testing of the complete system installed in its true operating environment in the field. However, there are several shortcomings and problems with this approach. The set of test cases will in practice never completely cover full system functionality and the set of system requirements, even if they according to the traceability documentation are claimed to do so. Consequently, testing will identify existence of errors, rather than proving absence of errors. Further, errors are (too) often discovered (too) late in the development cycle, making them time consuming and expensive to correct.

Errors in a safety critical system's requirements are very expensive to correct if not discovered before the system is installed in the field. The reason for this is that (software and/or hardware) modifications of safety critical systems implies an exhaustive verification and validation process to be performed before the modified system can be put into ordinary use. For this reason, it is for safety critical systems in particular very important to the largest possible extent to ensure the correctness of the system's requirements before the system is installed in the field. This can (with some limitations) be done with the use of formal methods.

Formal methods imply that a mathematical model of the system and its behavior is created. Applied on this mathematical model, formal analysis can show if the system's requirements are:

- complete (e.g., there are no system states where no, or ambiguous, behavior is specified);
- consistent (e.g., there are no requirements prescribing contradictory behavior);
- guaranteeing the system's specified attributes.

Furthermore, by selecting a suitable type of formal model (see below), the system can also be designed and implemented, possibly including automatic code generation, based on the same formal model. If so, the entire system development lifecycle from requirement creation to implementation with full verification and validation will then be based on the same formal model.

There are a number of formal methods which can be used for safety critical systems, each having different properties making them suitable for different types of applications. There is no single method being the best for all types of applications. Also, a lot of work has been done applying formal analyses to the different graphical diagrams of the widespread modelling tool UML [89]–[93].

Concerning how standards and regulations regard the use of formal methods, the EN50128:2011 standard classifies "Formal methods" as a "Highly Recommendable" technique to be used in e.g. the most safety critical (SIL3 and SIL4) railway applications. A number of different formal methods are also explicitly identified by the EN 50128:2011. However, despite its obvious advantages, applying a formal method has also its disadvantages:

- The nature of the application must be suitable for formal analysis. Most suitable are applications which can be expressed as combinatorial logic (e.g., ATM flight trajectory planning or a railway interlocking system), not suited are for instance asynchronous event communication systems.
• The use of formal methods requires competencies both of the application domain (e.g., ATM) as well as the formal method in itself, a combination which is hard to find.

Therefore, in practice, formal methods are most often used as a supplement to, and not a replacement for, traditional methods, and for limited parts of the total system. One could see how the critical and complex parts of the system is covered and verified by formal methods, while the outer regions are not.

Software development in general has undergone radical changes during the last couple of decades, with agile software development becoming the de-facto standard in many areas. Recent techniques like DevOps let operations take an active part in the process of not just maintaining a system, but also deciding on changes. While complete systems earlier relied most on custom-built hardware, solutions now more frequently include off-the-shelf hardware components with increasingly complex software on top. We see this trend also within development of safety-critical systems. The development process of such systems has, however, been the traditional waterfall model, or the V-model as described in D2.1. SafeScrum, an agile development methodology for safety-critical systems brings forward the idea that not only it is viable to develop and assess safety-critical software built this way, it will also reduce the cost and enhance the development and assessment process. This is possible due to agile development and its focus on communication between actors, and extensive use of tools for automation of documentation and tests.

Another trend we see in safety-critical systems is the shift in infrastructure, including the ATM domain. Systems are increasingly connected across unsecure networks, substituting physical separation with logical splits. Changes in network security could then possibly create unsafe conditions that need to be remedied, at the very least trigger an impact analysis. Making decisions about these concerns also depend on other variables, like cost and organizational decisions. Currently there exists no tool that maps these different challenges and balance concerns of various types.

9 www.safescrum.no
4 Requirements for the PACAS reasoning techniques

Table 2 summarizes the requirements for the PACAS reasoning techniques. We should emphasize that these constitute a preliminary set of requirements for the PACAS reasoning support. We enumerate each requirement of this first iteration, state the source of the requirement, and provide the description. There are five possible sources for each requirement, being in line with the process followed in D2.1 and D3.1:

1. Expertise: The partners’ expertise in projects within the ATM domain,
2. State-of-the-art: Techniques and practices that are proven to be successful,
3. Interview: The results of the interviews with the domain experts,
4. Gap: Identified gaps in the literature that are relevant to PACAS, as identified by the needs,
5. Decision: Project choices made by the consortium.

<table>
<thead>
<tr>
<th>ID</th>
<th>Source</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁</td>
<td>State-of-the-Art, Interview</td>
<td>PACAS reasoning techniques shall analyse strategic objectives from multiple perspectives.</td>
</tr>
<tr>
<td>R₂</td>
<td>State-of-the-Art</td>
<td>PACAS reasoning techniques shall be supported within each PACAS view (economic, organizational, security, safety).</td>
</tr>
<tr>
<td>R₃</td>
<td>State-of-the-Art, Gap</td>
<td>PACAS reasoning shall integrate different findings and support multi-objective analysis.</td>
</tr>
<tr>
<td>R₄</td>
<td>Gap</td>
<td>PACAS shall support multi-objective reasoning techniques for trade-off analysis.</td>
</tr>
<tr>
<td>R₅</td>
<td>Interview, Expertise</td>
<td>PACAS reasoning techniques shall be automated.</td>
</tr>
<tr>
<td>R₆</td>
<td>Decision, Expertise</td>
<td>The execution of PACAS reasoning techniques shall produce findings which will be visualized over the different models.</td>
</tr>
<tr>
<td>R</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>Expertise</td>
<td>PACAS automated reasoning shall be augmented with informative messages to explain findings, especially situations of conflicts in a given view, to facilitate the interaction of decision makers.</td>
</tr>
<tr>
<td>8</td>
<td>Gap</td>
<td>PACAS shall support the visualization of trade-off decision points to help decision makers not only to comprehend but also to compare the impact of changes.</td>
</tr>
<tr>
<td>9</td>
<td>State-of-the-Art, Decision</td>
<td>PACAS reasoning techniques shall identify the optimal solution in an efficient manner.</td>
</tr>
<tr>
<td>10</td>
<td>Interview, Decision</td>
<td>PACAS reasoning techniques shall make use of KPIs to validate decisions.</td>
</tr>
<tr>
<td>11</td>
<td>State-of-the-Art, Expertise</td>
<td>PACAS shall ensure that reasoning results are traceable wrt. decisions they contribute to.</td>
</tr>
<tr>
<td>12</td>
<td>Expertise</td>
<td>PACAS automated reasoning shall take into account measures of significance related to each change, covering e.g., complexity, reversibility, novelty needed in implementation, etc.</td>
</tr>
<tr>
<td>13</td>
<td>Expertise</td>
<td>PACAS reasoning techniques shall be useful to both traditional development processes as well as newer agile methodologies.</td>
</tr>
<tr>
<td>14</td>
<td>Gap</td>
<td>PACAS reasoning techniques shall be linked to concepts such as views, operational improvements steps, SESAR solutions and enablers that are part of EATMA.</td>
</tr>
<tr>
<td>15</td>
<td>Gap, Decision</td>
<td>PACAS automated reasoning shall rely on semantic relatedness techniques to identify overlaps and discrepancies between multiple views.</td>
</tr>
</tbody>
</table>

R7 has emerged during the interviews from the need to have any change deeply analysed and from multiple perspectives, see Section 3.2. This requirement is one that the PACAS consortium had identified from the state-of-the-art analysis at the time of writing the proposal. It is an essential requirement to the nature of the solution proposed by the PACAS project.

R8 follows on the need to deal with change and impact propagation first within each PACAS perspective, see Section 2.1. Before understanding the repercussions of change in the other perspectives, it is important to first check whether such change affects the initial perspective.

R9 builds on the need to integrate the different findings to support multi-objective analysis, putting together the results of reasoning over the different PACAS perspectives. This requirement also dates back to the PACAS proposal, and is identified as a gap from the literature (see Section 2.4).
Putting together the findings is not enough. As a consequence, the reasoning techniques will support finding a trade-off solution that fine-tunes the different objectives related to the different PACAS perspectives, i.e., related to security, safety, economic and organizational concerns. This is identified as a gap, for current approaches need to be extended to support finding an optimal solution for change management in the ATM domain. This is $R_4$.

Reasoning techniques are not necessarily automated. However, in complex systems as ATM the need for automation is crucial to support the identification of decision points and help experts in their work. This is a known problem to our consortium, but it also emerged during the interaction with AB members, therefore $R_5$ stems from interviews and our expertise.

Producing findings and offering automated support offer only a partial solution to facilitating the work of the expert if these are not complemented with a visual aid and textual messages where applicable. Therefore, $R_6$ and $R_7$ cover these aspects, being a result of a decision taken by the consortium and based on our expertise in the matter. $R_8$ is in line with these two requirements, with the difference that it is specialized for the visualisation of decision points as part of the trade-off analysis, which is a gap.

The PACAS consortium requires that the reasoning techniques find the optimal solution in an efficient way, that is why it has decided that the reasoning techniques for trade-off analysis shall be built upon optimization modulo theories, given that they are used in state of art techniques in reasoning with multiple objectives. As such, $R_9$ is important not to reinvent the wheel, while making advancement with respect to current related work.

KPIs are important indicators not only in economics, but also to validate decisions. This emerged during the interviews with AB members and will be considered for the development of reasoning techniques.

$R_{11}$, $R_{12}$, and $R_{13}$ are derived from the discussion in Section 3.3 of the needs from past experience. Traceability of automated reasoning results is important in order to know on what basis a particular decision has been made ($R_{11}$), which is also a requirement originating from the software safety standards including Regulation 482/2008 and ED-12/DO-178. It is further useful for the reasoning to take into account not only the impact the change has on the system, but also the overall undertaking and any uncertainties related to implementing the change ($R_{12}$). While PACAS should be suited for use in traditional development processes, agile methodologies should not be excluded from using PACAS due to their growing use in also safety critical systems development ($R_{13}$).

Finally, $R_{14}$ and $R_{15}$ were identified as needs at the first face-to-face interaction with project officers. These requirements are important in order to bridge different modelling frameworks and relate to the projects that are part of EATMA ($R_{14}$), as well as to take advantage of the already available knowledge structures and taxonomies ($R_{15}$).
5 Conclusions

This deliverable makes three contributions. First, we present the state of the art for the domains that span the problem space of the PACAS project. Second, we elicit preliminary user needs for the PACAS reasoning techniques by conducting interviews and interacting with the AB members during the 1st validation workshop in Rome. Third, we propose a set of requirements based-on: (i) our expertise in the ATM domain and our analysis; (ii) gaps in the state of the art that we identify as relevant to the PACAS project; (iii) decisions we have made based on needs that we wanted to be more specific about due to potential constraints. The latter is quite relevant given the preliminary input received from the interviews related to the background of the interviewees.

There are still a number of open questions related to the PACAS reasoning support, which will be investigated as the project moves forward. It is important to note that requirements are described at a high level due to the difficulty for AB members to know what they want the reasoning techniques to be, in advance, especially with the modelling language being under development. Our next steps will ensure to take this into account, including defining precisely what needs to be satisfied, and how this shall be done.

We will follow an iterative and incremental process to provide different versions of the reasoning techniques that will be developed alongside the PACAS modelling approach and modelling language(s).
References


[31] E. Paja, A. Mat, C. Woo, and J. Mylopoulos, “Can Goal Reasoning Techniques be used for Strategic Decision-Making?”


[61] “CENELEC - Standards Development - List of Technical Bodies -.”


[63] “IEC Functional Safety and IEC 61508.”


[91] X. He, “Formalizing UML class diagrams-a hierarchical predicate transition net approach,” in
