First release of the reasoning proof -of- concept

D4.2
PACAS
Grant: 699306
Call: H2020-SESAR-2015-1
Topic: Sesar-10-2015 ATM Architecture
Consortium coordinator: UNITN
Edition date: 30 May 2017
Edition: 00.01.00
Authoring & Approval

Authors of the document

<table>
<thead>
<tr>
<th>Name/Beneficiary</th>
<th>Position/Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabiano Dalpiaz/UUTR</td>
<td>WP2 Leader</td>
</tr>
<tr>
<td>Fatma Başak Aydemir/UUTR</td>
<td>Project member</td>
</tr>
<tr>
<td>Martina Ragosta/DBL</td>
<td>WP5/6 Leader</td>
</tr>
<tr>
<td>Mattia Salnitri/UNITN</td>
<td>Project Member</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>Elisa Chiarani</td>
<td>Project assistant</td>
<td>19/05/2017 + 29/05/2017</td>
</tr>
<tr>
<td>Erlend Andreas Gjære</td>
<td>WP3 Leader</td>
<td>18/05/2017 + 24/05/2017</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/UNITN</td>
<td>Project coordinator</td>
<td>30/05/2017</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>11/04/2017</td>
<td>Draft</td>
<td>Fatma Başak Aydemir, Fabiano Dalpiaz</td>
<td>Proposed table of contents</td>
</tr>
<tr>
<td>00.00.02</td>
<td>12/04/2017</td>
<td>Draft</td>
<td>Fabiano Dalpiaz</td>
<td>Reviewed table of contents</td>
</tr>
<tr>
<td>Date</td>
<td>Action Description</td>
<td>Author(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------</td>
<td>----------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13/04/2017</td>
<td>Reviewed table of contents</td>
<td>Martina Ragosta</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20/04/2017</td>
<td>Inserted Section 2</td>
<td>Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30/04/2017</td>
<td>Inserted section 3</td>
<td>Fabiano Dalpiaz, Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/05/2017</td>
<td>Inserted Section 4</td>
<td>Fabiano Dalpiaz, Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/05/2017</td>
<td>Inserted Section 5</td>
<td>Fabiano Dalpiaz, Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/05/2017</td>
<td>Inserted executive summary and the abstract</td>
<td>Fabiano Dalpiaz, Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/05/2017</td>
<td>Reviewed and modified document</td>
<td>Fabiano Dalpiaz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14/05/2017</td>
<td>Applied comments</td>
<td>Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18/05/2017</td>
<td>First Internal Review</td>
<td>Erlend Andreas Gjære</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19/05/2017</td>
<td>Quality check</td>
<td>Elisa Chiarani</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19/05/2017</td>
<td>Internal Review</td>
<td>Mattia Salnitri</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22/05/2017</td>
<td>Implemented the requests from the reviews</td>
<td>Fabiano Dalpiaz, Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24/05/2017</td>
<td>Doc. Ready for the 2nd internal review</td>
<td>Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24/05/2017</td>
<td>Final Internal Review</td>
<td>Erlend Andreas Gjære</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26/05/2017</td>
<td>Implemented the comments from the review</td>
<td>Fatma Başak Aydemir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29/05/2017</td>
<td>Final quality check</td>
<td>Elisa Chiarani</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30/05/2017</td>
<td>Approval for submission</td>
<td>Paolo Giorgini</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PACAS

PARTICIPATORY ARCHITECTURAL CHANGE MANAGEMENT IN ATM SYSTEMS

This deliverable has received funding from the SESAR Joint Undertaking under grant agreement No 699306 under the European Union’s Horizon 2020 research and innovation programme.

Abstract

The main objective of PACAS WP4 is to provide solutions based on automated reasoning for the conceptual modelling and decision-making processes included in the PACAS Change Management process. In this deliverable, we report on our efforts on building the Intelligent Cross-view Alignment (ICVA) service for the PACAS platform, which provides an analysis of models that is based on natural language processing (NLP). The ICVA service suggests missing concepts to a model from a common air traffic management vocabulary based on the concepts present in other models. The deliverable also presents two other reasoning mechanisms to be implemented for the project: (i) an NLP-based dashboard for the modellers and (ii) a multi-objective reasoning solution to help the decision making process to select the final solution.
## Table of Contents

**Executive summary** .................................................................................................................. 6

1. **Introduction** .......................................................................................................................... 7
   1.1 PACAS Overview .................................................................................................................. 7
   1.2 Relationship with other deliverables .................................................................................... 8
   1.3 Structure of this document .................................................................................................... 8

2 **Baseline for Intelligent Cross-view Alignment** .................................................................... 9
   2.1 Multi-view Modelling ......................................................................................................... 9
   2.2 Natural Language Processing ............................................................................................. 10

3 **Intelligent Cross-view Alignment** ........................................................................................ 13
   3.1 Overview ............................................................................................................................ 13
   3.2 Heuristics ............................................................................................................................ 18
   3.3 Implementation and Integration with the Platform ............................................................... 20
   3.4 Evaluation ........................................................................................................................... 23

4 **Ongoing and Future Work** ................................................................................................... 25
   4.1 Visualization of modelled concepts ..................................................................................... 25
   4.2 Multi-objective reasoning for solutions .............................................................................. 26

5 **Conclusions** ......................................................................................................................... 30

**References** .................................................................................................................................. 32

Appendix 1: PACAS requirements identified for reasoning techniques ................................. 34

Appendix 2: Material for the evaluation session during the 2nd validation workshop ............ 36
Executive summary

PACAS promotes a participatory change management process for Air Traffic Management (ATM) systems. The platform is designed to facilitate discussions over conceptual models to base arguments on solid evidence from the models itself or the analysis results derived from the models. The models are built by experts with various expertise, working for diverse organizations located in all over Europe. To reliably compare these models and analysis results, the experts should ensure that the models cover the same parts of the domain. This deliverable provides the first release of a solution for this challenge.

Our Intelligent Cross-View Alignment (ICVA) service provides a Natural Language Processing (NLP) based analysis of the models, which provides concepts from the Air Traffic Reference Model (AIRM) as suggestions to PACAS modellers. AIRM has a key role as a unifying standard for semantic interoperability within European ATM systems. The benefits of this approach are twofold. First, the modellers are made aware of the concepts modelled in other models. Second, since the suggestions are from AIRM which is intended as a common vocabulary for European ATM systems, they are encouraged to use this common terminology to prevent ambiguity or misunderstandings in between models. The implementation details, integration with the main PACAS platform, and the results of the evaluation session during the 2nd validation workshop are provided.

This deliverable also explains two other reasoning mechanisms for the PACAS platform that will be developed during the second half of the project. The first solution includes an NLP-based analysis of the models to discover common and missing concepts among models, as well as possible ambiguities. The results are going to be presented in forms of Venn diagrams and the solution is going to be integrated to the main platform as a dashboard for the modellers. The second solution is a multi-objective reasoning support to discover the optimal solution based on an objective function, that blends quantitative and qualitative objectives. We also report the progress on the requirements identified for the reasoning mechanisms at the beginning of the project in Appendix 1.
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.

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, organisational, security, and safety concerns (Figure 1).

1.2 Relationship with other deliverables

This document relates to several other deliverables of the PACAS project. The deliverables on the consolidation of the state-of-the-art and gap analysis of the project clearly influenced the first design of the reasoning solutions (D4.1 [1]). Furthermore, deliverables related to platform design and releases (D2.1 [2] and D2.2 [3]) set the requirements and constraints for the reasoning. Implementation and integration of the reasoning solutions requires strong collaboration with the platform implementation, so the work presented in this deliverable was in parallel with the work that led to D2.3 [3]. The reasoning techniques presented in this deliverable are applied to models created with the languages chosen in WP3; as such, there are clear ties with D3.1 [4] and D3.2 [5]. Finally, the validation plan devised in D5.1 [6] is at the basis of the evaluation activities reported in this deliverable.

1.3 Structure of this document

The rest of the document is structured as follows:

- Section 2 presents the related work from the areas of multi-view modelling and natural language processing for the reasoning service developed and integrated with the platform.
- Section 3 details the NLP-based reasoning service to support the inter-model alignment in the PACAS platform.
- Section 4 describes two additional reasoning support for the PACAS platform as future work.
- Section 5 concludes the document.
- Appendix 1: PACAS requirements identified for reasoning techniques reports the progress for the requirements identified for the reasoning mechanisms at the beginning of the project.
- Appendix 2: Material for the evaluation session during the 2nd validation workshop includes the material prepared for and presented at the evaluation session during the second validation workshop.
2 Baseline for Intelligent Cross-view Alignment

The sections of this chapter present the baseline for the automated reasoning techniques that are presented in this deliverable: multi-view modelling for representing multiple perspectives, and natural language processing techniques that enable understanding text and labels. The former section provides an overview of the existing tools and techniques that offer multi-view modelling, which have not been covered in D4.1. The latter section summarizes the information provided in D4.1 and enriches with the libraries and toolkits for practical applications of theory.

2.1 Multi-view Modelling

Conceptual models are produced and read by humans. As the complexity of the system being modelled increases, the models become more complex as well, and the modelling process requires the collaboration of multiple modellers as supported in the PACAS project. However, collaboration introduces some additional challenges such as conflict resolution, for the modellers are likely to have their own perspectives that are not necessarily consistent [7]. In this section, we provide an overview of the literature that supports collaborative modelling via multiple views, additional information specific for modelling enterprise architectures and modelling languages for individual PACAS views is present in D3.1. At the end of the section we briefly compare PACAS and existing methods.

The terms view and viewpoint are used interchangeably in the literature. We follow the definition of Fischer et al. [8] and define a viewpoint as a language which represents a meta-model. A viewpoint could be a subset of the original meta-model, or a stakeholder may restrict herself to use only the concepts and relationships of her interest. A view on the other hand is an instance of a viewpoint, it presents a model. Given that a meta-model provides a language for a model, a viewpoint is the subset of the language provided by the original meta-model, whereas a view is a model created by using this subset.

Nicolaescu et al. [9] formalize and implement a web-based, multi-view collaborative modelling framework by separating the meta-model and the visualizing elements formally, and letting users customizing the visual elements for the concepts and relationships that are included in the meta-model. Users can also collaboratively define viewpoints (subsets of the meta-model) that they use in their views. The tool demonstrates its features in i* [10] goal models that represent stakeholders’ requirements and social dependencies.
Sirius\(^1\) is a framework built on top of Eclipse Modeling and Graphical Modeling Frameworks (EMF and GMF) and supports creating modelling tools in a model-driven fashion. It also allows specifying custom viewpoints and visual styles. Despite the relative ease of use, it is a desktop application and does not offer any features to support collaboration among modellers.

MetaEdit+ [11] is a toolset to define modelling languages (meta-modelling) and generate model editors. It offers collaborative view and viewpoint editing; however, the collaboration is not real-time for there is a lock mechanism that prevents it.

Multiple views are considered for Object Oriented Databases (OODBs). MultiView [12] is a popular view extension for Gemstone OODB [13]. CVG-Algorithm introduced by MultiView enables users defining viewpoints. The algorithm automatically adds classes and relationships to a viewpoint when a reference to an object or relationship is included to the meta-model.

Existing literature on multi-view modelling assumes that there is one common meta-model from which viewpoints (and then views) are created. In PACAS, however, there are different aspects of a change issue to be investigated, and the four perspectives supported by the project use different meta-models. There is no single meta-model that unites the meta-models for the individual views. Stakeholders are able to share their views using different languages to support their arguments in the platform. However, this flexibility makes it difficult to track misalignments and detect conflicts among models.

One possible way of overcoming this challenge is to create a unified meta-model. However, this solution is not feasible, as different stakeholders may prefer different languages; four languages chosen for the project just demonstrate the potential of the proposal. Unifying different meta-models is also difficult since one concept may be possibly mapped into many others for the languages are not necessarily Domain Specific Languages (DSLs).

To overcome these challenges, we need methods to (i.) automatically detect the content of the models, (ii.) support an aligned co-evolution of the views, and later (iii.) compare solutions provided by the models and discover the optimal solutions. In the next section, we overview the potential of NLP techniques to provide support for modellers for the first two of these three tasks.

### 2.2 Natural Language Processing

A conceptual model contains selected information about the domain that is modelled. Types of elements and relationships is one source of information. Another source is the labels of elements and

\(^1\) [https://www.eclipse.org/sirius/features.html](https://www.eclipse.org/sirius/features.html)
relationships. Consider a goal labelled “performance increased” in a goal model. The type of the element indicates that it is a desired state of affairs. The label indicates that the desired state is increased performance. Various kinds of analyses can be conducted based on the types of elements and relationships. D4.1 covers a selected set of such analyses. Other useful information can be extracted by analysing the labels of the documents. Labels include text in natural languages and we argue that automated reasoning techniques for conceptual modelling can benefit from analysing the text of the labels.

Natural Language Processing (NLP) concerns between computers and natural languages and has many sub-areas such as automatic summarization, sentiment analysis, relationship extraction, question answering, topic discovery, and so on. We aim to apply NLP techniques to identify the content of the conceptual models (in our case, ATM concepts modelled in the models). In the same fashion, NLP-supported automated reasoning techniques may be applied to conceptual models to discover relationships, or answer questions about the models.

Vectors representing the distributional similarity information of words are input for many practical NLP applications. These vectors are trained using the word2vec\(^2\) \([14]\) family of algorithms. Using such vectors, it is possible to assign similarity scores to word tuples, retrieve the most similar words for a given word, or form analogies. For example, subtracting the vector of “France” from the vector of “Paris” and adding the vector of “Italy” returns a vector similar to that of “Rome”.

\[
\text{Vector(Paris)} - \text{Vector(France)} + \text{Vector(Italy)} = \text{Vector (Rome)}
\]

One can also retrieve synonyms (the word with the same meanings), hyponyms (sub-types of a word), or hypernyms (parent types of a word) from lexical databases of natural languages. WordNet \([15]\) is such a database for English and it is used as a source by many toolkits.

<table>
<thead>
<tr>
<th>Word</th>
<th>flight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyponyms</td>
<td>Acrobatics, flyover, glide, pass, solo, sortie, terrain flight, ...</td>
</tr>
<tr>
<td>Hypernyms</td>
<td>Air travel, aviation, air</td>
</tr>
</tbody>
</table>

\(^2\) https://code.google.com/archive/p/word2vec/
Semantic similarity of two words is a metric defined, where two words are closer (distant) if they have similar meanings (not similar meanings). Sense2vec\(^3\) analysis for “flight” on Reddit\(^4\) data from 2015 returns [plane, flights, nearest airport, airplane, takeoff, emergency landing, shuttle, LAX, flight home, layover, return trip, ...]. Semantic similarity results also depend on the training corpora for a given algorithm.

Lexical semantic relatedness, on the other hand, is not only concerned about whether the concepts have similar meanings or close to each other in the hyponymy taxonomy. A popular example is the tuple “spaghetti-tomato sauce” where “spaghetti” and “tomato” are not similar, but definitely related (which can be discovered analysing the co-occurrence of the concepts in cashier receipts, for example).

NLTK [16] is a natural language processing toolkit written in python. It provides interfaces to various corpora and data sets including WordNet, and offers functionalities to parse sentences, categorize text, and analyse linguistic structure. TextBlob\(^5\) is another text processing library that is built upon NLTK. Its features include noun phrase extraction, sentiment analysis, tokenization and others. Another popular library is spaCy\(^6\) which is intended for fast statistical analysis of text. It also has an integrated visualizer, displaCy\(^7\), to visualize named entities in a text.

---

\(^3\) https://demos.explosion.ai/sense2vec/?word=flight&sense=auto  
\(^4\) https://www.reddit.com  
\(^5\) https://textblob.readthedocs.io/en/dev/  
\(^6\) https://spacy.io  
\(^7\) https://demos.explosion.ai/displacy-ent/
3 Intelligent Cross-view Alignment

3.1 Overview

The Intelligent Cross-view Alignment (ICVA) service is a part of the PACAS Project that aims to assist the modellers to collaborate as they work on the same change issue. As detailed in deliverables D2.2 and D2.3, a change issue is a high level topic of discussion whose result may lead to a change in an ATM system. A change issue is divided into decision points, where experts collectively identify the as-is situations and then later possible solutions to those situations.

PACAS acknowledges the power and necessity of the team work, and encourage collaboration within Europe by suggesting a work process that involves teams. Teams may be formed by grouping experts according to different criteria:

- Shared area of expertise (e.g., security team, safety team): this way, experts from different institutions discuss a change issue in depth with respect to their common expertise.
- Country or region, where the experts with different expertise discuss a change issue holistically from the perspective of their country or region.

The PACAS platform supports discussions not only as comment threads but also through conceptual models where experts support or refute arguments via the models they create.

One challenge is to make sure that the teams capture the same (chunk of the) domain so that the experts can do reliable comparisons in between models and analysis results. The diversity of modelling languages used for modelling the ATM domain makes it unfeasible to attempt creating an overarching meta-model that would enable us to directly mapping modelling constructs from one language to the other. First, the languages may have different levels of granularity, so a mapping between a domain specific language to a general purpose language or vice versa does not always provide a useful transformation. Second, the meta-model has to be constantly maintained to keep it up-to-date with the new languages that are used. Our conclusion is that the benefits of this maintenance does not justify the effort and there are more effective methods supported by artificial intelligence that yield similar if not more effective solutions for aligning different models in the PACAS platform.

One such method is to employ NLP. NLP methods have been used to extract and aggregate information, to summarize text, and to answer questions. Inspired by the applications of the NLP on regular text, news articles, and social media posts, we apply NLP to conceptual models that are created and shared in the PACAS platform. Conceptual models also include various text as labels of
the model elements and relations, as descriptions of elements, and as notes for the overall model. The main difference between the text used in conceptual models and the regular text, such as news articles, is that in conceptual models there usually is a lack of full sentences. Labels often include short descriptions and abbreviations due to the limited visual space. We develop techniques to overcome these challenges when analysing the text in conceptual models to align them.

Our proposal is to suggest concepts from a common reference to the modellers based on the content of the models that are created and shared for the same change issue. First, we keep track of the modelling process. As the modellers submit significant changes of the models they work on, we analyse the models to extract the nouns and calculate the nouns frequencies to understand the semantics of the model over the nouns used in the labels of the model elements. We update the data on the models during the modelling process and use these data to provide suggestions. When a modeller requests suggestions or the platform automatically communicates some suggestions, the ICVA service compares the current state of the model with the other models that model the same change issue and identify the missing nouns. Based on the list of the missing nouns, the ICVA service discovers the related concepts from a common reference and use them send them to the model for consideration. Upon receiving the suggestions, we leave the decision of including them to the model to the modeller herself, the ICVA service does not automatically modify any models. One reason behind this choice is that we are not capable of accurately locate and connect new model elements using automated tools. The second reason is that conceptual modelling is a process requires human intelligence and creativity, and not one that can be fully automated. Automatically modifying the models might confuse the modeller, and correcting the output of the automated reasoning might require more time than modelling from scratch. Figure 2 summarizes the main steps of the ICVA service. Below we provide the details of each step.

Figure 2: Overview of the ICVA service

**Step 1:** For this version of the ICVA service, we limit our scope to the nouns used in the models. For the future versions, we may consider more elaborate analysis including adjectives and verbs. Analysing nouns is a basic analysis since models capture a domain over “things” that exist in that domain.

Each time a model is significantly modified, that is, a commit has been made by the modeller, the ICVA service receives the commit information from the PACAS platform which includes the added, deleted, and modified elements. We analyse each label and extract all the nouns from the labels.
We then identify the frequency of a noun in a model as an indication of how much a model is about (related to, talks about) that noun. So we calculate the frequency of each noun in the model and update the existing noun frequency data we keep about the model.

**Step 2:** Once it is time to send suggestions to a PACAS model, we first identify nouns that are not included in that specific model. Next, we use Air Traffic Reference Model (AIRM) [17] to discover the exact domain-specific terminology related to the nouns. AIRM is a common reference for information and data models developed for European ATM models. We use this model to discover the exact terminology related to the nouns that are modelled in other PACAS models, but not present in the model that asks for suggestions.

**Step 3:** Once the modeller receives suggestions from the ICVA service, she is asked for her feedback. We are aware that the feedback of the modeller can be in a broad spectrum, whether the concept is related to the model that she is modelling, whether she considers the concept useful. Whether or not she includes the model in the future is also an implicit feedback about the suggested concept.

![Figure 3: The pilot provides some suggestions to the modeller](image)

Our aim is not to overwhelm the modeller with questions about the suggestions, hence we only collect feedback whether she finds the suggestion useful or not. If the concept is not found useful, we do not suggest it again. Similarly, if a suggested concept is later included in the model, it is not sent to the modeller as a suggestion (per Step 2). The user interface for the suggestions is also key for being nonobtrusive to the modeller, yet its design is beyond the scope of the ICVA service and related to the platform [3], [3].
Figure 4 includes an exemplary part from a safety view modelled using fault trees concerning sector unification in different flight levels. The labels extracted from this model and nouns extracted from the labels are in Table 2:

Table 2: Nouns contained in the labels of Figure 4

<table>
<thead>
<tr>
<th>Label</th>
<th>Nouns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased airspace complexity, unpredictability</td>
<td>['airspace', 'complexity', 'unpredictability']</td>
</tr>
<tr>
<td>Compromised safety</td>
<td>['safety']</td>
</tr>
<tr>
<td>Flights on cruise</td>
<td>['Flights', 'cruise']</td>
</tr>
<tr>
<td>Flights transiting Germany only</td>
<td>['Flights', 'Germany']</td>
</tr>
<tr>
<td>Increased number of separation manoeuvres</td>
<td>['number', 'separation', 'manoeuvres']</td>
</tr>
<tr>
<td>Huge variety of vertical manoeuvers</td>
<td>['variety', 'manoeuvres']</td>
</tr>
</tbody>
</table>

All nouns but two occur only once. The nouns “flights” and “manoeuvres” occur twice. The frequency information may be discarded for this toy model. If a modeler requests suggestions for an empty model, the ICVA service searches AIRM for related concepts and return the results. We need further heuristics to select and limit the number of results for too many suggestions may result in decision fatigue for the modeler. One example is related to the noun “flight”. There are 88 concepts only in the Information Model of AIRM where the string “flight” occurs. This number increases where the search is conducted also in class descriptions. We discuss heuristics for finding related concepts and filtering them in the next section.
Table 3 Concepts form AIRM Information Model whose names contain ”flight”

<table>
<thead>
<tr>
<th>FlightEnvelope</th>
<th>ExtendedFlightPlan</th>
<th>FlightCrewMember</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlightControlUnit</td>
<td>CommercialFlightPlan</td>
<td>FlightRules</td>
</tr>
<tr>
<td>FlightManagementSystem</td>
<td>FlightPlan</td>
<td>PoliceFlight</td>
</tr>
<tr>
<td>FlightConfiguration</td>
<td>CurrentFlightPlan</td>
<td>FlightLevelChange</td>
</tr>
<tr>
<td>UpperFlightInformationRegion</td>
<td>FlightObjectDataset</td>
<td>FlightTypeChange</td>
</tr>
<tr>
<td>FlightInformationRegion</td>
<td>WhatIfFlightObjectDataset</td>
<td>FlightRulesChange</td>
</tr>
<tr>
<td>FlightPathAlignmentPoint</td>
<td>RepetitiveFlightPlan</td>
<td>ICAOFlightID</td>
</tr>
<tr>
<td>AircraftFlightStatus</td>
<td>FiledFlightPlan</td>
<td>IATAUniqueFlightIdentifier</td>
</tr>
<tr>
<td>FlightPriority</td>
<td>FlightScript</td>
<td>FlightDesignatorSuffix</td>
</tr>
<tr>
<td>OperationalFlightPlan</td>
<td>AirFiledFlightPlan</td>
<td>GloballyUniqueFlightIdentifier</td>
</tr>
<tr>
<td>RequestedFlightLevel</td>
<td>FlightIntent</td>
<td>OriginFlightDate</td>
</tr>
<tr>
<td>ReclearanceInFlight</td>
<td>MilitaryFlightActivity</td>
<td>FlightDesignator</td>
</tr>
<tr>
<td>OperatorFlightPriority</td>
<td>FlightSequence</td>
<td>IATAFlightNumber</td>
</tr>
<tr>
<td>FlightPlanning</td>
<td>FlightLevel</td>
<td>LandingAndTakeoffFlightPhases</td>
</tr>
<tr>
<td>PostFlight</td>
<td>FlightPhaseEfficiency</td>
<td>FlightPhase</td>
</tr>
<tr>
<td>FlightExecution</td>
<td>HorizontalFlightEfficiencyAssessmentOutput</td>
<td>DirectFlightSegment</td>
</tr>
<tr>
<td>FlightConditionCombination</td>
<td>HorizontalFlightEfficiencySubjectOfAssessment</td>
<td>FlightIntent</td>
</tr>
<tr>
<td>FlightRestrictionLevel</td>
<td>FlightPhaseEfficiencyAssessment</td>
<td>FlightPlannedRoute</td>
</tr>
<tr>
<td>FlightRestriction</td>
<td>HorizontalFlightEfficiencyAssessment</td>
<td>FlightDeck</td>
</tr>
<tr>
<td>FlightRoutingElement</td>
<td>HorizontalFlightEfficiencyAssessmentCriterion</td>
<td>FlightCrewApplicationAndApproval</td>
</tr>
<tr>
<td>FlightRestricitonRoute</td>
<td>FormationFlight</td>
<td>OperationalFlightInformationService</td>
</tr>
<tr>
<td>FlightSuspension</td>
<td>InstrumentFlightRules</td>
<td>FlightInformationService</td>
</tr>
<tr>
<td>CurrentTacticalFlightModel</td>
<td>SpecialRequirementsFlight</td>
<td>FlightSafetyDocumentsSystem</td>
</tr>
<tr>
<td>FiledTacticalFlightModel</td>
<td>Flight</td>
<td>FlightInformationCentre</td>
</tr>
<tr>
<td>RegulatedTacticalFlightModel</td>
<td>MilitaryFlight</td>
<td>FlightDataManagerPublisher</td>
</tr>
</tbody>
</table>
The ICVA service is a language-agnostic approach, which fits well into PACAS project. STS-ml [18], fault trees [19], and BIM [20] are used to capture security, safety, and organizational perspectives, respectively. The ICVA service does not only work with these three languages, but it can also be integrated to other modelling languages and diagrams such as BPMN, or UML class diagrams as long as the labels of the model elements can be extracted. In particular, the ICVA service can be integrated with state-of-the-art enterprise architecture modelling platforms such as the MEGA Enterprise Architecture tool that is used in EATMA8.

### 3.2 Heuristics

The ICVA service relies on a combination of several heuristics for the suggestions to the modeller. In this section we discuss these heuristics.

**Stemming**: Nouns can be used in multiple forms in a model, such as singular or plural. Stemming can be useful to unify the different instances so the noun frequencies can be accurately aggregated. However, stemming may also result in over-aggregation when multiple nouns are derived from a single stem. Consider the examples provided below:

- Example word list: ['flight', 'removal', 'altitude', 'coordinates', 'instructions', 'situations']
- Stemmed list: ['flight', 'remov', 'altitud', 'coordin', 'instruct', 'situat']

Stemming libraries do not stop after singularization but continue to stem all postfixes appended to the stem of the word. In this case multiple words that are derived from the same stem are reduced back to that stem, and the differences are lost. However, omitting stemming results in counting ‘conflict’ and ‘conflicts’ as separate words, and a non-accurate frequency count for the model.

**Noun phrases**: A noun phrase is consisted of two or more nouns, such as “flight envelope”. When nouns are extracted from labels, noun phrase information is lost, so the modelled words are kept as “flight” and “envelope” and not “flight envelope”. One side effect of such approach is the

frequencies of words that commonly occur in noun phrases increase more. For example, having “flight level” (2 times), “flight plan” (1 time), and “flight state” (1 time) phrases in a model increases the frequency of “flight” more than the other words in the noun phrases and AIRM will be queried for “flight” which is the most frequent word and not for “flight level” which is the most frequent noun phrase. However, if the model that requests suggestions include the word “flight”, then other words will be used to query AIRM and the corresponding classes will be suggested. We are still experimenting with noun phrases and their usage. One obstacle is that there is still room for improvement for extracting noun phrases from the text. Due to the nature of element labels (incomplete sentences), it is even harder to detect noun phrases in the labels. As a result, the ICVA service’s default option is the noun extraction rather than noun phrase extraction.

No-space spelling in AIRM is not a real challenge as we can easily remove the space in strings of noun phrases when we query AIRM.

Exact match vs similarity vs relatedness: Once the nouns that are present in other models but missing in the model for which the ICVA service provides suggestions, AIRM is queried to get concepts that are relevant for that noun. The querying can be done in several ways.

- Get an exact match in the class name: the class name from the AIRM should be an exact match for the noun we query. Although it is possible to get a match for nouns such as “conflict”, this option is the most restrictive option in terms of results.
- Get an exact match in the name or in the description: in AIRM each class has also a description attribute which provides a longer explanation for the class. For example, the class “AirTrafficOperations” does not have the string flight in its name but in its description, “Information on all ATM-related activities required to implement air traffic by enabling flights.”
- The class name contains the noun: any class that contains the noun in its name is returned. This query returns more results especially when stemming is used.
- The class name or the description contains the noun: any class that contains the noun in its name or in its description is returned.
- The class name is synonym of the noun: the class name that is the synonym of the noun is returned.
- The class name or the description contains the synonym of the noun: the class that contains the synonym of the noun in its name or in the description is returned. For example, when the word “aim” occurs in a model, its synonym “intent” may result in the suggestion “FlightIntent”.
- Classes that have high relatedness score: classes whose names have high relatedness scores with the noun.
- Classes that contain words that have high relatedness score in their name or description: classes that contain one or more words that have high relatedness score with the noun. Relatedness score depends on the training data. For example, “disallow” in the model may have a high relatedness score with “restriction” which later results in “FlightRestriction”.

Parent and child classes in AIRM: Once a class is identified as a suggestion using any of the above heuristics, the suggestion list can be enhanced by returning the parent and/or the child nodes.
Depending on the preferences of the users, parent classes provide a broader perspective whereas the child classes add specific details to the models. Currently, we experiment with such preferences. In future release the platform can provide an interface to users to let them specify their preferences over the heuristics. In the distant future, machine learning techniques can be employed to learn the preferences of the users from their feedback.

**Fostering focus vs creativity:** The goal of the heuristics above is to find classes from AIRM that are related to the missing nouns in various ways. Suggestions based on such queries foster focus and alignment with other models. Another direction is to foster creativity by providing suggestions that are not exact matches, or related classes but classes \( x \) steps further away from the pinpointed classes where \( x \) is a creativity constant and each step is a relation in AIRM. So if ‘flight’ is identified as a related class, another concept, such as “VolcanicAsh” that is 5 steps away from flight is provided as a suggestion to the user.

**How many suggestions?** Too many suggestions may overwhelm the modeller, and too few suggestions may not have the intended impact on the modelling process. Screen space is also an important factor for determining the number of suggestions. The number of suggestions can be taken as input from each modeller, or can be learned over time. Based on our interviews with the advisory board members, the current number of suggestions is set to 5.

### 3.3 Implementation and Integration with the Platform

The ICVA service is implemented in Python\(^9\) due to the abundance of NLP libraries. The web service is created using the Django web framework\(^10\). Django organizes the content of the web application under three categories: models, views, and templates (MVT).

Models: each model in Django is a single, definitive source of information from data. In PACAS, primitive models are the following:

- **PACAS Model:** model identification number (integer), change issue identification number (integer), decision point identification number (integer) are stored in this model.
- **PACAS Concept:** concept identification number (integer) and concept name (varchar) are stored in this model.
- **ModelConcept:** this model keeps data about which concept (Modeled_concept (foreign key from PACAS Concept)), is model in which model (Modeled_in (foreign key from PACAS Model)) with which frequency (frequency (integer)).

---

\(^9\) [https://www.python.org/](https://www.python.org/)

\(^10\) [https://www.djangoproject.com/](https://www.djangoproject.com/)
- PACAS Suggestion: suggestion identification number (integer), suggested AIRM concept (varchar) and suggested_to_model (foreign key from PACAS Model) are the fields of this model.
- PACAS Feedback: feedback identification number (integer), suggestion (foreign key from PACAS Suggestion), timestamp (time), is_useful (boolean) are the information stored in this model.

The Django framework provides database APIs for SQLite\textsuperscript{11}, MySQL\textsuperscript{12}, and PostgreSQL\textsuperscript{13}. The lightweight structure of SQLite fits well with the database needs of the ICVA service, so it is the selected database framework of choice. Updating and querying the database tables are done via the Django APIs.

Views: a view is a callable which takes a request and returns a response. There are three main views of the ICVA service.

- Commit: The main PACAS platform posts the commit information as a JSON object to the ICVA service when the modeller commits a model. The JSON object includes the commit operation (add, delete, modify), element ID (a unique string for each element in the model), and the element label. The ICVA service process the data and updates the database accordingly. If the databases are successfully updated the view returns a HTTPResponseSuccess. For caller methods other than POST, the view returns HTTPResponseBadRequest.
- Suggestions: currently the PACAs platform is implemented in a way that the user specifically asks for suggestions herself by clicking the ‘Get Suggestions’ menu item in the modelling pane. When there is such request, the PACAs platform calls the service via POST method, and sends over the current state of the model as a JSON object. Once the ICVA service receives the JSON object
  1. the database tables are updated to reflect the current modelled elements and their frequencies;
  2. missing concepts are identified by comparing the current state of the model with the modelled concepts in the database;
  3. AIRM is queried for suggestions based on the most frequent missing items using a combination of the heuristics listed in the previous section, received suggestions are filtered based on feedback;

\textsuperscript{11} https://www.sqlite.org/
\textsuperscript{12} https://www.mysql.com/
\textsuperscript{13} https://www.postgresql.org/
4. suggestions are sent to the PACAS platform as a JSON object.

Similar to the commit view, the ICVA service returns HTTPResponseSuccess if the suggestions are successfully created. If the view is called any method other than POST, the ICVA service returns HTTPResponseBadRequest.

- Feedback: once a modeller receives the suggestions and provides her feedback on them, the main PACAS platform sends the feedback in JSON format to the ICVA service via POST method. The ICVA service updates the database and use the feedback for future calls for filtering suggestions to be provided.

Apart from the main views, create model view is called when a new PACAS model is created in the main PACAS platform so that the information about the model is entered to the ICVA database. Similarly, a delete model view is called when a PACAS model is deleted in the main platform. For the test and development instances of the service there is also a view to reset the database by deleting all data.

The URL structure for the views are http://ICVA-DOMAIN/PARAMETERS/VIEW-NAME. Currently, the ICVA service is hosted at http://pacas.science.uu.nl:8080/service and the only parameter needed is the identification of the model. So a commit for PACAS model 21 is sent to the following url: http://pacas.science.uu.nl/21/commit

In this sense, the ICVA service can easily be integrated to other modelling platforms as it uses the standard protocols and exchange data with JSON objects. As a security measure, currently it can only be accessed from the allowed hosts.

Figure 5: A security expert who created an STS-ml model using the PACAS platform is shown suggestions (bottom-left)
The end-user interacts with the ICVA service through the main PACAS platform. A commit action commits the model to the platform yet also the database of the ICVA service is updated. The ‘Get suggestions’ menu item is in the main menu of the platform. Suggestions are delivered via the “pilot”, the avatar that is located at the bottom left of the modelling screen as can be seen in Figure 5 and Figure 3. In this sense, the ICVA service also adds intelligence to the gamification elements of the platform. The user feedback on suggestions are also collected via the co-pilot, so it is a way for the user to engage with the platform in a gamified manner.

Templates: templates are used to dynamically generate HTML source. As the ICVA service does not present data on the browser, we do not use any templates.

3.4 Evaluation

We have presented an offline version of the ICVA service during the 2nd validation workshop in Rome, Italy on 04-04-2017. At the time the ICVA service was not fully integrated with the main PACAS platform, so models and suggestions are presented to the subjects on paper. We then collected the feedback from the participants about the suggestions. The first objective of the evaluation was to analyse the ICVA service suggestions for the purpose of evaluation with respect to their relevance to the commit from the point of view of AB Members in the context of multi-perspective modelling of the ATM domain. The second objective was to analyse the tags suggested by the ICVA service for the purpose of evaluation with respect to their usefulness to the corresponding perspective from the point of view of AB Members in the context of multi-perspective modelling of the ATM domain.

The evaluation plan was as following:

1. Explain how the ICVA service works to the participants
2. Present the initial safety model (Figure 13) and then the latest commit to the model (Figure 14).
3. Provide a short explanation of the models and allow subjects to get familiar with them.
4. Present the initial security model (Figure 11).
5. Provide a short explanation of the model and allow subjects to get familiar with it.
6. Present the suggestions to security model based on the safety model.
7. Present the final security model (Figure 12).
8. Present the suggestions to safety model based on the security model.
9. Users fill the questionnaire
10. Discussion with the AB members

The evaluation is conducted with two groups during one hour sessions. The first group had three participants whereas the second group had 2 participants.

Results: Four out of five participants agreed that the ICVA service helps to improve the quality of models (Q1 and Q3 of the questionnaire provided in Appendix), one participant disagrees. Three out of five participants agree that the ICVA service helps to speed up the modelling process to the security view, one participant neither agrees nor disagrees and one participant disagrees (Q2). Two
out of five participants agree that the ICVA service helps to speed up the modelling process for the safety model, two neither agree nor disagree and one participants disagrees.

**Discussion:** One of the participants disagreed that the ICVA service helps to improve the quality of the models and speed up the modelling process. During the discussion he explained his position based on the selected aspects security and safety for the demonstrations. He stated that such models would not feed each other during the modelling process in ATM, rather the output of the security model will feed the safety model. So his comments were about the selected aspects but not on how well the ICVA service works.

Open ended questions got mixed replies. Some of the participants found high level suggestions such as Trajectory relevant to the model, whereas the others preferred lower level suggestions such as TrajectoryChangePoint. The same applies to the usefulness of suggested points. We used this feedback to develop the heuristics detailed in Section 3.2. Future experimentation (both within PACAS e beyond the scope of the project) is necessary to assess what are the most effective heuristics.

During the discussion, we gathered additional feedback and requirements related to when and how to present the suggestions to the modeller, both of which has an impact on the perceived usefulness of the suggestions. The subjects had a variety of preferences on the matter. Two of the participants preferred to receive the suggestions at the beginning of the modelling session when they have fresh ideas. Other two argued that they would be focused on their model at the beginning of the modelling process so they preferred having suggestions after some time they start modelling. As a result, the platform had a menu item through which the modellers manually request suggestions. In the future, with further customization options automatically presenting suggestions could be seamlessly integrated into the platform. The participants also had diverse opinions on the visualization of the suggestions. They all preferred five to seven suggestions on the screen through the avatar, yet some requested a suggestion dashboard where they can check the complete list. The others preferred the suggestions presented in the speech bubble to be clickable so that they can explore higher and lower level suggestions through a tree-like visualization. The PACAS platform should experiment with different visualization techniques to accommodate such requests.
4 Ongoing and Future Work

4.1 Visualization of modelled concepts

In order to fulfil the request for a dashboard specifically for the modellers by the advisory board members during the second validation workshop, we have drafted a solution for visualizing the common and distinct concepts in different models.

The visualization is in the shape of a Venn diagram, where modelled concepts are shown as elements within the boundaries of the model they belong. Common concepts are shown in the intersection of the models. Moreover, the ambiguity among some concepts are detected, and highlighted as a warning to modellers towards using a shared vocabulary. Figure 6 depicts the overall visualization whereas Figure 7 provides a close-up on the visualization of the concepts.
van der Schalk [21] et al. identify potential ambiguities and missing requirements from sets of user stories and visualize them in a fashion like Figure 6. They use the classification proposed by Shaw and Gaines [22] to classify state of concepts when comparing two models.

<table>
<thead>
<tr>
<th>Distinction</th>
<th>Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>Consensus</td>
</tr>
<tr>
<td>Different</td>
<td>Conflict</td>
</tr>
</tbody>
</table>

Table 4: Classification of concept states by Shaw and Gaines [22]

Our plan is to use the work of van der Schalk et al. as the baseline for our dashboard and adapt it to the needs of the stakeholders of the PACAS project so that the modellers are clearly able to see similar or potentially ambiguous concepts as well as the missing concepts. The input format will be different from user stories for there are conceptual models in the PACAS project. Also, there might be more than 3 models to compare and contrast, which could make the visualization challenging.

Figure 7: A close-up from the early adaptation of [21] to PACAS

### 4.2 Multi-objective reasoning for solutions

Our plan is to utilize multi-objective reasoning (MOR) to suggest a final decision at the end of PACAS change management process for each solution based on how well the solutions (i) satisfy the relevant key performance indicators (KPIs) or (ii) contribute differently to the various objectives. This kind of automated processing is intended to help resolve the inherent divergences that the decision makers will be confronted with due to the existence of multiple viewpoints (e.g., a solution that is very good for security may not be equally good for safety) and organizations.
The former case (satisfaction of KPIs) is demonstrated in Figure 8 and is turned into a quantitative MOR problem, whereas the latter case (contribution to objectives) is presented in Figure 9 and is solved through qualitative analysis. Our plan is to combine the qualitative and quantitative analysis techniques to let PACAS stakeholders freely express the contribution of the solutions to the objectives and discover the optimal solutions based on their objective function and objective models.
Below we provide an example of the quantitative optimization problem whose setup is described above. In the example depicted in Table 5, there are three enumerated solutions, each of which is given a score for three objectives: cost, security, and safety. If the objectives are evaluated individually, the solution that costs the least is Solution 1. Solution 2 is the best (optimal) solution for the safety objective and Solution 3 is the optimal solution for security. The problem is to decide which one is the best solution when all objectives are considered. The first step towards a solution is to construct an objective function. Assuming that the objectives are to minimize cost and maximize security and safety scores, a simple objective function is the following in which all objectives have the same weight, so they are considered equally important.\(^{14}\)

\[
O(x) = 0.33 \cdot v_{\text{safety}} + 0.33 \cdot v_{\text{security}} - 0.33 \cdot v_{\text{cost}}
\]

Then the objective scores are calculated as \(O(S1) = 56.397, O(S2) = 55.704,\) and \(O(S3) = 55.671.\) \(S1\) has the highest objective score so it is the optimal solution. In real scenarios in the ATM domain, we expect to have many more objectives. As a result, assigning accurate objective weights and then discovering the optimal solution is more challenging.

<table>
<thead>
<tr>
<th></th>
<th>Solution 1</th>
<th>Solution 2</th>
<th>Solution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost (Million Euro)</td>
<td>4.1</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td>Safety Score (0-100)</td>
<td>87</td>
<td>92</td>
<td>91</td>
</tr>
<tr>
<td>Security Score (0-100)</td>
<td>88</td>
<td>81</td>
<td>91</td>
</tr>
</tbody>
</table>

In case of qualitative representation of the problem, the decision makers may not specify the amount of the contribution of the solutions to the objectives, or they may use discrete increments such as −, +, ++ to represent very negative to very positive contributions. In Figure 9, – and + symbols are used to indicate a positive or a negative contribution. The advantage of qualitatively expressing the contribution is that it is easier to represent the contribution, so time and effort are saved since there is no need to conduct analyses to quantify contributions. Another advantage is that an objective hierarchy might be represented, in the style of goal models.

\(^{14}\) The cost, safety, and security values should be normalized to get accurate results this such objective functions
Figure 10 is a toy example for a qualitative objective problem. If there is a known contribution from a solution to an objective, the contribution is represented via an edge, and it is labelled with a + or – sign depending on the positive or the negative impact. It is relatively easier to represent the contributions, and the techniques might be implemented even when there are missing contributions. Objectives may also be prioritized. For the example below, if “ATCO workload reduced” has higher priority than “automation level increased” Solution 3 can be selected as the optimal solution.

![Figure 10 An example qualitative multi-objective optimization problem](image)

Our goal is to devise reasoning techniques that can accommodate both quantitative and qualitative reasoning as well as a combination of the two. The reasoning technique to be devised is highly related to the representation method of the problem, which will be reported in the future deliverables of WP3.
5 Conclusions

This deliverable presents an automated reasoning service prototype that is also integrated with the main PACAS platform presented in D2.3. The main goal of the ICVA service is to assist the modellers of the platform to model similar if not the same concepts to ensure the models are aligned, which would later make them reliable sources to compare and contrast different situations and solution proposals. The ICVA service was introduced to and evaluated by the AB members during the 2nd validation workshop.

There are several benefits of the ICVA service being an independent web service. First, it frees the service and the client from the restrictions on programming languages and frameworks. In this case, the PACAS main platform is implemented using Java and JSP whereas the ICVA service is implemented in python using different tools. Second, data exchange over JSON objects enables a standard way of non-restrictive communication among various platforms. Third, the same reasoning service can serve to multiple clients (e.g., enterprise architecture tools) without the need of sharing data or integration among the client platforms.

The ICVA service applies NLP techniques to conceptual models, which is an innovative approach for aligning conceptual models without the need for mapping the meta-models. This provides flexibility for models modelled in different languages can easily benefit from the reasoning service. In this sense, the ICVA service is language-agnostic while it relies on a domain-specific taxonomy of terms (SESAR’s AIRM).

The ICVA service received a mostly positive evaluation and raised a healthy discussion during the 2nd validation workshop. The original idea was found to be potentially useful and the suggestions on the models provided during the evaluation sessions were found to be related to the models. The evaluation session also revealed the need for different heuristics based on different preferences of the AB members. Further fine tuning and experimentation are required for the system to reach its optimal performance.

Future work includes two other types of reasoning mechanisms for the PACAS platform. The first one is the visualization of common and distinct concepts among PACAS models to support modellers through the modelling process. The goal is similar to the ICVA service’s goal but the visualization is intended to present the state of the models rather than having a set of suggestions based on AIRM. The modellers can consult the visualization to check whether there are some concepts missing in their own models or whether ambiguous terminology is used among different models.

Multi-objective reasoning support for identifying the optimal solutions is also part of the future work. Our plan is to let the PACAS stakeholders to express the contribution of solutions on different objectives which may also have a hierarchy among themselves and use quantitative and qualitative
multi-objective reasoning techniques to discover the optimal solutions based on the objective function.
References


## Appendix 1: PACAS requirements identified for reasoning techniques

Below we present the table of requirements that are presented in D2.2 and report on their status.

<table>
<thead>
<tr>
<th>ID &amp; Title</th>
<th>Status</th>
<th>Covered in</th>
</tr>
</thead>
<tbody>
<tr>
<td>R4.1 PACAS reasoning techniques shall analyse strategic objectives from multiple perspectives.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.2 PACAS reasoning techniques shall be supported within each PACAS view (economic, organizational, security, safety).</td>
<td>In progress</td>
<td>Sections 3 and 4</td>
</tr>
<tr>
<td>R4.3 PACAS reasoning shall integrate different findings and support multi-objective analysis.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.4 PACAS shall support multi-objective reasoning techniques for trade-off analysis.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.5 PACAS reasoning techniques shall be automated.</td>
<td>In progress</td>
<td>Section 3 and 4</td>
</tr>
<tr>
<td>R4.6 The execution of PACAS reasoning techniques shall produce findings which will be visualized over the different models.</td>
<td>In progress</td>
<td>Section 4.1</td>
</tr>
<tr>
<td>R4.7 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>
<td>In progress</td>
<td>Section 3</td>
</tr>
<tr>
<td>R4.8 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>
<td>In progress</td>
<td>Section 3 and 4</td>
</tr>
<tr>
<td>R4.9 PACAS reasoning techniques shall identify the optimal solution in an efficient manner.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.10 PACAS reasoning techniques shall make use of KPIs to validate decisions.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.11 PACAS shall ensure that reasoning results are traceable wrt. decisions they contribute to.</td>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>R4.12 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>
<td>In progress</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>Requirement</td>
<td>Status</td>
<td>Section</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>R4.13 PACAS reasoning techniques shall be useful to both traditional development processes as well as newer agile methodologies.</td>
<td>Out of scope</td>
<td></td>
</tr>
<tr>
<td>R4.14 PACAS reasoning techniques shall be linked to concepts such as views, operational improvements steps, SESAR solutions and enablers that are part of EATMA.</td>
<td>In progress</td>
<td>Section 3 and 4</td>
</tr>
</tbody>
</table>
Figure 11 Initial security model
Figure 12 Final security model
Figure 13 Initial safety model
Suggestions to security model based on the safety model

- Trajectory
  - TrajectoryChangePoint
  - TrajectoryConstraint
  - TrajectoryPoint
- Assignment
  - SlotAssignmentHistory
- Distribution
  - DistributionInformation
- Overload
  - Overdelivery
  - Imbalance
- EmergencyOperations
- DemandAndCapacityBalancing
- Conflict

Suggestions to safety model based on the security model

- Flight Level
  - Exit Flight Level
PACAS Intelligent CrossView Alignment User Questionnaire

Q1: The concepts suggested to the security view are relevant to the security view.
   1. Strongly agree
   2. Agree
   3. Neither agree nor disagree
   4. Disagree
   5. Strongly disagree

Q2: The concepts suggested to the security view are useful to the modeller.
   1. Strongly agree
   2. Agree
   3. Neither agree nor disagree
   4. Disagree
   5. Strongly disagree

Q3: The concepts suggested to the security view can improve the quality of the security view.
   1. Strongly agree
   2. Agree
   3. Neither agree nor disagree
   4. Disagree
   5. Strongly disagree

Q4*: The concepts suggested to the security view can speed up the modelling process.
   1. Strongly agree
   2. Agree
   3. Neither agree nor disagree
   4. Disagree
   5. Strongly disagree

Q5: Among the concepts that are suggested to the security view, which concepts are the most relevant to the security view and why?

-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
Q6: Among the concepts that are suggested to the security view, which concepts are the least relevant to the security view and why?

Q7: Among the concepts that are suggested to the security view, which concepts are the most useful to the security view and why?

Q8: Among the concepts that are suggested to the security view, which concepts are the least useful to the security view and why?

Q9: Based on the safety models you have seen, what other concepts would you suggest to the security view? Why?