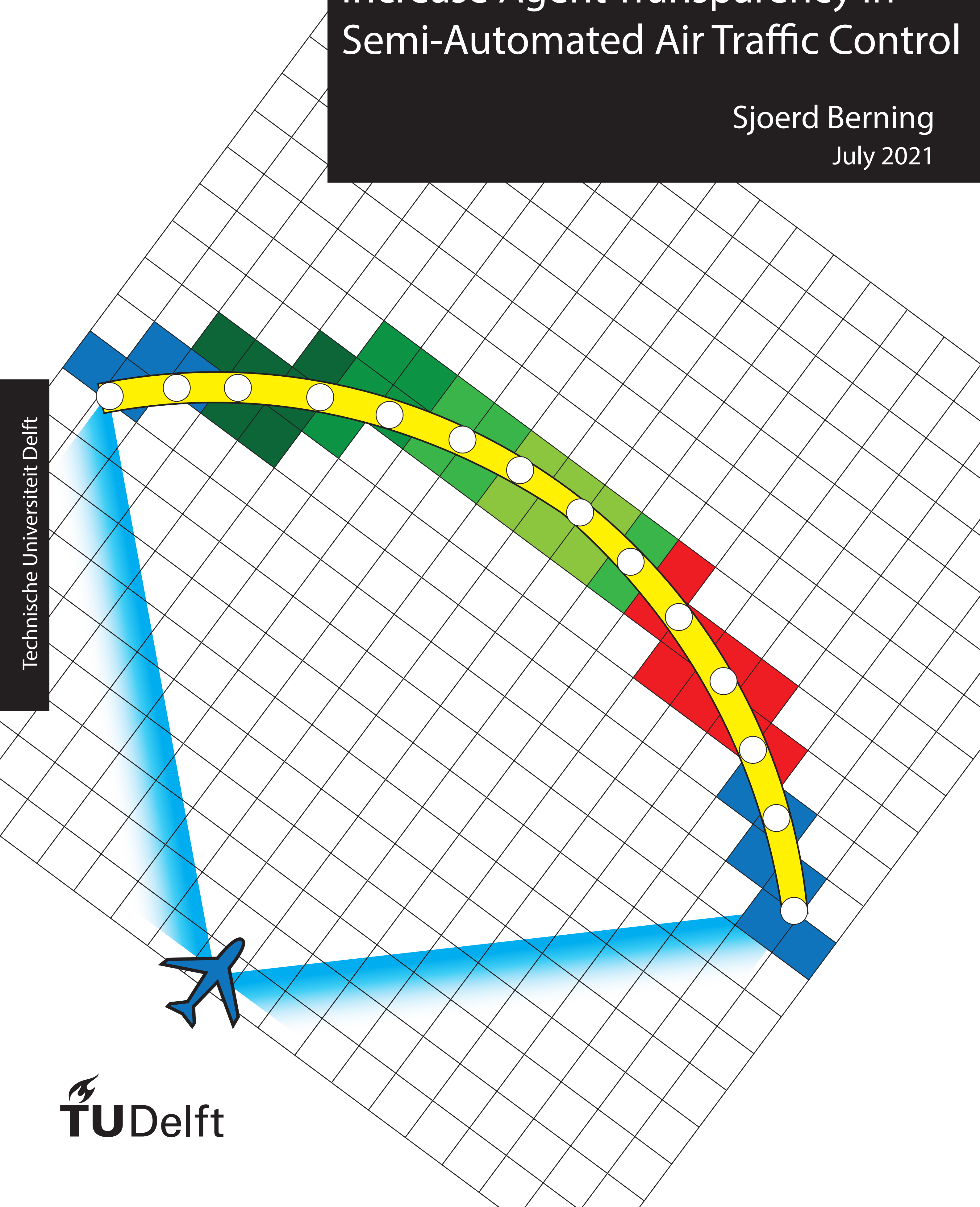
Ecological Approach to Increase Agent Transparency in Semi-Automated Air Traffic Control

Sjoerd Berning

July 2021



Technische Universiteit Delft

TU Delft

Ecological Approach to Increase Agent Transparency in Semi-Automated Air Traffic Control

by

S. Berning

Student number:	4370589	
Project duration:	September 10, 2020 – July 7, 2021	
Thesis committee:	Prof. dr. ir. M. Mulder,	TU Delft, Chair
	Dr. ir. C. Borst,	TU Delft, Supervisor
	Dr. ir. B.F. Lopes dos Santos	TU Delft, Examiner
	Ir. G. de Rooij	TU Delft
	Ir. T. Monteiro Nunes	TU Delft

TU Delft

Preface

In the beginning of June 2020, my thesis supervisor, Clark, told me about this opportunity to do a master thesis in the area of explainable automation. I was immediately hooked because of its overlap between many disciplines (psychology, aviation, programming) and because of the importance of making algorithms and automation explainable in many areas of society.

In September, I started this master project and after 10 months of hard work I am happy to present to you my master thesis titled: *"Ecological Approach to Increase Agent Transparency in Semi-Automated Air Traffic Control"*. The thesis was conducted at the Control and Simulations group of the Aerospace Faculty at the Technical University of Delft.

Even though the process was not always easy due to the COVID-19 Pandemic situation, I consider this project as one of the most valuable learning experiences during my studies at the TU Delft. An important element for this was the support I received from my thesis supervisors Clark, Gijs, Tiago and Max. Thank you for the new ideas, critical mindset, endless support in programming and for familiarizing me with doing research. Additionally, I would like to thank all participants that volunteered to be part of the experiment.

Furthermore, I want to thank Daan and Michiel for the many (online) work sessions and opportunities to present, test and discuss ideas. Svenja, thanks for endlessly reading my report for any mistakes, but more importantly, thanks for your love and moral support. Finally, I would like to thank my family, especially my parents, for a wonderful upbringing that brought me where I am today, but also for being a solid base from which I could and can discover the world. Extraordinary!

S. Berning
Leiden, June 2021

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Description**
ACC	Area Control
ACOD	Aircraft Conflict Detection Tool
AH	Abstraction Hierarchy
APP	Approach Control
ATC	Air Traffic Control
ATCO	Air Traffic Control Operator
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
CEL	Grid Scenario
CON	Confidence
CPDLC	Controller Pilot Data Link Communication
CTR	Control Zone
CTA	Control Areas
CWA	Cognitive Work Analysis
EID	Ecological Interface Design
FL	Flight level
FIR	Flight Information Region
HCA	Human Centered Automation
HMI	Human Machine Interaction
HR	Hit Ratio
INT	Time Interval Scenario
ICAO	International Civil Aviation Organisation
KBB	Knowledge Based Behaviour
LOA	Level of automation
LOS	Loss of separation
LVNL	Luchtverkeersleiding Nederland
LSM	Largest Separation Margin
MAHALO	Modern ATM via Human/Automation Learning Optimisation
MTCD	Midterm Conflict Detection
MUAC	Maastricht Upper Area Control Center
RADAR	Radio Detection and Ranging
RTA	Required Time of Arrival
RBB	Rule Based Behaviour
SBB	Skill Based Behaviour
SA	Situation Awareness
SAT	Situation Awareness-Based Agent Transparency
SEP	Separation (Blockage) Scenario
SES	Single European Sky
SESAR	Single European Sky ATM Research

Abbreviation	**Description**
SP	Shortest Path
SRK	Skill Rule Knowledge
SSD	Solution Space Diagram
STCA	Short Term Conflict Alert
STR	Cost Scenario
TCT	Tactical Controller Tool
TIM	Time Scenario
TMA	Terminal Control Area
TSR	Travel Space Representation
Q1	Question 1
Q2	Question 2
Q3	Question 3
Q4	Question 4
UTA	Upper Airspace

I

Thesis paper

Ecological Approach to Increase Agent Transparency in Semi-Automated Air Traffic Control

Sjoerd Berning
Supervisors: Clark Borst, Gijs de Rooij, Tiago Monteiro Nunes and Max Mulder

Future ATM systems will rely on automation to make operations more efficient. Creating insight into the inner-workings of automation, also known as agent transparency, is expected to play an important role for effective human-machine collaboration. This research proposes an ecological approach to increase agent transparency in automated rerouting for en-route traffic. For the purpose of this study, an ecological interface for the rerouting task, developed in a previous study, was visually augmented with the constraints guiding the behavior of an experimental path-planning algorithm. This was done in two different ways: a top-down and bottom-up approach. The top-down approach starts at the goal of the system and subsequently adds information related to the physical implications, while the bottom-up approach has the reversed order. The design was tested in a human-in-the-loop experiment with ten participants. Results show that higher levels of transparency significantly increased actual and perceived understanding of the agent's decisions. Furthermore, the top-down approach performed significantly better in questions related to the strategy of automation, while the bottom-up approach was found more useful for making predictions about the agent's rationale for making certain decisions. Future research should investigate how agent and domain transparency could be combined and should test situation awareness in addition to understanding of automation. Additionally, because only static situations were investigated in this study, the effects of a dynamic work domain featuring various time-critical situations should be analyzed in future research.

Index Terms—**Agent Transparency, Air Traffic Control, Ecological Interface Design, Understanding, Explainability, Human-Machine, Decision Support Systems.**

I. INTRODUCTION

THE current Air Traffic Management (ATM) system needs to increase its effectiveness and efficiency in order to keep up with demand. The SESAR ATM Master Plan [1] states that an increase in automation is required and can lead to lower costs, better environmental protection, more efficient use of airspace and a reduction of Air Traffic Control Operators (ATCOs) workload. Nevertheless, the Master Plan also prescribes that the human operator will keep its central position in the ATC system; at least for the foreseeable future. Therefore, humans and machines need to collaborate at the core task of the ATCO, which is the rerouting task. This task ensures separation through management of airspace disruption like traffic conflicts and adverse weather.

Issues may arise due to the implementation of new automation. Bainbridge [2] formulated the ironies of automation, which indicate ways in which problems related to human-automation interactions may increase rather than decrease as a result of the increased level of automation. One irony states that knowledge of a system is best obtained by active participation. However, automation is pushing humans towards a more passive supervisory role resulting in a more difficult task for humans to maintain their understanding of the system.

Furthermore, additional layers of automation make the system more complex and difficult to understand, leading to more problems when the human is forced to intervene. Automation can then be seen as a black box [3]. A glass box [4], in which the human operator can understand the inner-workings of automation, can help out in this human-automation issue. Changing the black box into a glass box can be seen as increasing agent transparency [5].

Transparency can have many benefits. Unclear reasoning by the automation results in the operator questioning the automation's accuracy and effectiveness [6]. Transparency can be a solution for this [7]–[9]. Eiband et al. [10] suggest that transparency can be a means to increase the user's mental model of the system as a whole, which in turn increases trust, control effectiveness and user satisfaction [11]–[13]. Furthermore, transparency could be used in addition to strategic conformance [14] or could (partially) replace it. It is expected that adding transparency to the system will result in the opportunity to make the system less conformal, which could lead to more optimal solutions [15]. Finally, according to Mittelstadt et al. [16], transparency is required to allow meaningful oversight by the human. They state that human intervention becomes increasingly difficult if the human has less information than the machine. This comes down to a call for transparency to increase trust and understanding as well as control over automation [17].

It still remains a challenge how agent transparency can be best achieved. One approach is to focus on the agent. The Situation Awareness Agent Based Transparency (SAT) model [7] describes three levels that together tackle different aspects of the agent. They state that agent transparency is the ability of the interface to convey the agent's intent, performance, future plans and reasoning process to the human operator. By doing so, this approach focuses on fitting the interface to the algorithm with the risk of presenting too much (unnecessary) information. The SAT model is good for determining types of visualization, but lacks handles for selecting the right information. In this research, an ecological approach is suggested in which the work domain forms the starting point. The effect of the agent upon the work domain is analyzed and during interface design both the constraints of the work domain and the agent are visualized. Thereby, this approach is hypothesised to create the necessary handles to select the right information of the agent. Furthermore, the ecological basis can be used to support human operators in unanticipated events [18]. The ecological approach tries to fit the algorithm to the interface instead of the other way around.

In this research, the Travel Space Representation (TSR) [19], an ecological interface for the rerouting task, was used as the ecological basis and the Flexibility Metric algorithm [20]–[22] was chosen as the agent that automated this task. The research objective is to propose a design for an ecological ATC interface that improves understanding of automated conflict resolution in perturbation management by designing and testing interface elements that visualize the crucial steps of the Flexibility Metric algorithm. The interface elements are tested by means of a human-in-the-loop experiment with ten participants. The aim is to investigate whether higher levels of transparency create a better understanding of automation. Furthermore, it is investigated if the human operator can be best assisted by bottom-up (starting from agent constraints) or top-down (starting from the solution) reasoning support.

This article has the following structure. Section II discusses agent transparency and how it can be created. The rerouting task, Flexibility Metric and TSR are discussed in Section III. In Section IV, the design of the interface is presented. This section starts with the work domain analysis and then discusses all elements of the interface. Additionally, it is stated how the interface can support the human in their task of understanding automation. The approach of the experiment is presented in Section V. This is followed by the results of this experiment in Section VI, which are discussed in Section VII. Finally, Section VIII concludes this research.

II. BACKGROUND: AGENT TRANSPARENCY

A. *Domain and Agent Transparency*

Transparency is a broad concept and is used across many fields of research. For this research an important distinction is made between domain transparency and agent transparency.

Domain transparency aims to portray the laws of physics that govern the work domain. This leads to a common ground or shared representation of understanding for both automation and operator [18]. Ecological Interface Design (EID) [23] can be used to create domain transparency as it aims to reveal the deep structure (constraints and relationships) of the work domain. The shared representation can be used for effective human-machine collaboration as it provides the required insights into automation's rationale as well as insights into the physical laws tied to the problem [24]. Therefore, the ecological basis remains crucial even if the rerouting task is completely automated. If the automation fails, ATCOs still need insight into the work domain to take over control effectively. Additionally, domain transparency gives context to the automation's chosen solution by portraying the physical laws. A dyadic approach that focuses primarily on the agent and/or human might simplify or supplant this crucial information making supervision and intervention more difficult [24].

Agent transparency can supplement domain transparency by adding information about the reasoning process of the agent. The following definition for agent transparency is used: [15, p.2] "*the extent to which aspects of the agent's inner process underlying a solution can be observed and explained in human terms.*" Therefore, by creating agent transparency, the human operator should be able to understand how the agent is interpreting the work domain. Furthermore, the constraints that the agent applies to work domain should be made salient. This can be achieved by showing how the agent affects the work domain.

Figure 1 shows the levels of automation at four stages of automation [25] for the setting used in this research. The agent, Flexibility Metric, has a high level of automation at the *decision selection* stage of automation as its goal is to produce safe and efficient reroutes. The ecological basis, TSR, provides little information regarding this stage, because it shows the physical boundaries of the control problem, but does not show how automation is interpreting them. Therefore, agent transparency has to provide insight into how the agent is making a decision in order to make all three stages of automation transparent. Note that in this research the focus is on how the agent comes to a particular solution and therefore the implementation of the action is out of scope.

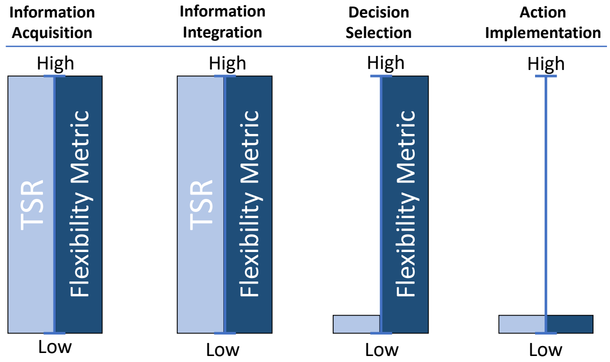


Fig. 1: Level of automation for the TSR and the Flexibility Metric at the four stages of automation.

B. *Implementation of Agent Transparency*

The Cognitive Work Analysis (CWA) [26] as part of the EID process can help in determining what information is relevant and how it should be presented. Furthermore, research in the field of transparency and explainability can help make clear how agent transparency can be implemented. A brief description of relevant concepts for constructing explanations is provided in this subsection.

Miller [27] states that explanatory algorithms can build upon research performed in the social sciences. An important finding states that human explanations are contrastive. People do not wonder why event X happened, but are more interested in why event X happened instead of event Y. Explanations are needed in response to a counterfactual case. Furthermore, explanations are a social process. They are transfers of knowledge in the form of a conversation or interaction.

Counterfactual and "why-not" explanations can be used to make contrastive explanations. Adadi et al. [28] describe counterfactual explanations as explanations that show the minimum condition that would have led to an alternative solution. Therefore, these type of explanations have a contrastive nature as they are a comparison between (at least) two options. Counterfactual explanations facilitate understanding of the sensitivity of the proposed solution, because they give insight about when the solution would change.

Another way to create contrastive explanations is via "why-not" explanations. These explanations allow for exploration of the field of possibilities. "Why-not" explanations are focused on explaining a certain anomaly the human operator might detect. Lim et al. [29] found that "why-not" explanations significantly increase understanding, trust and task performance.

Bhaskara et al. [5] address the issue that operators might get overwhelmed by the extra information given by transparent agents. This will make it more difficult to absorb all information and make decisions. Additionally, too much information could be harmful to the usage, trust and performance of the system [30] and could eventually lead to rejection of automation [31]. Springer et al. [31] propose the use of progressive disclosure. This theory states that explanatory information or increased transparency should only be provided if the human operator is in need of it. The concept of progressive disclosure can be created in an interactive manner. This is in accordance with Miller [27] and Abdul et al. [17] who state the importance of interactive explanations.

Both top-down and bottom-up processing can help operators in understanding the situation and agent [7]. This relates to Rasmussen's Abstraction Hierarchy (AH) [32] as the top-down approach starts at higher levels of abstraction and then moves to lower levels of abstraction, while the bottom-up approach has the reversed order. Bottom-up approaches are better suited to the technology itself as they start from the building blocks of the technology towards the goal of the system. In top-down processing, the goal of the system is the starting point and from there the physical implications are found. Ecological interfaces support primarily top-down processing while more technology driven display designs make use of a bottom-up approach [23]. In problem-solving situations, ecological interfaces support reasoning starting at higher levels of abstraction towards the more physical basis. Top-down processing is seen as a more efficient manner of problem-solving and more human oriented [23], but requires prior knowledge for effective usage.

C. *Measuring Operator Understanding*

By providing agent transparency, the idea is that understanding of the agent by the human operator can be increased. The three levels of Situation Awareness (SA) as described by Endsley et al. [33] can be used to indicate understanding. Selkowitz et al. [34] showed that SA Level 2 and 3 were significantly higher for higher levels of transparency. Wright et al. [35] found that higher transparency levels resulted in significantly higher scores of Level 1 SA. However, no significant effect was measured at Level 2 and 3 SA. Both researches measured the SA levels by queries related to the three levels of SA.

In order to measure the actual understanding of the agent, the queries can be designed such that they cover all relevant aspects of the agent as stated by the three levels of the SAT model [7]. This model is based upon the Endsley's Model [33] of SA, which describes the requirements for different levels of global SA. If all three levels of the SAT model are supported, the human operator is equipped to understand the agent and, if necessary, intervene.

The first SAT level provides the operator with the goals and planned actions of the agent. This level is about what the agent is trying to achieve and how it wants to achieve this. The second level supports the operator by communicating the reasoning process of automation by showing the constraints that the agent is considering. This level shows why automation is choosing a certain action. The third level is about projection of the agent's possible future states.

Another aspect of understanding is the perceived understanding of the operator. This can be measured by indicating the confidence level of the answers given to queries [36]. Automation is well calibrated if both perceived and actual understanding match. For example, it is not desired that operators are very confident about an answer while it is completely wrong or vice versa. Ideally, the perceived and actual accuracy should match one another.

III. SETTING

Creating agent transparency using an ecological approach is done in a particular context. This research focuses on the rerouting task of ATC and uses the TSR as ecological basis. The Flexibility Metric algorithm has been chosen as the agent that performs the rerouting task. All three concepts are described in more detail in this section.

A. *Rerouting task*

In this research, the effect of agent transparency is investigated in the context of perturbation management in en-route ATC. In this setting, an aircraft is flying through an airspace sector and due to some perturbation, for example a conflict with another aircraft, its current trajectory is not valid anymore as it would lead to a Loss of Separation (LoS). An overview of the rerouting task can be found in Figure 2. Human operators can reroute aircraft in such a situation by redirecting it to an intermediate waypoint such as point A in the figure. Speed and altitude are considered constant in this research.

B. *Travel Space Representation*

The Travel Space Representation (TSR) [19] is an ecological interface that aids human operators with performing the rerouting-task. The TSR is able to support the human by showing the feasible space for intermediate waypoint selection. By creating insights into the physical constraints tied to the problem, the TSR establishes a common ground between automation and the human operator.

An adapted version of the TSR forms the ecological starting point for achieving agent transparency. The TSR is adapted because speed is considered constant in this research. Therefore, the TSR will not be bounded to the 4D trajectory management, which defines the aircraft trajectory in space and time. The

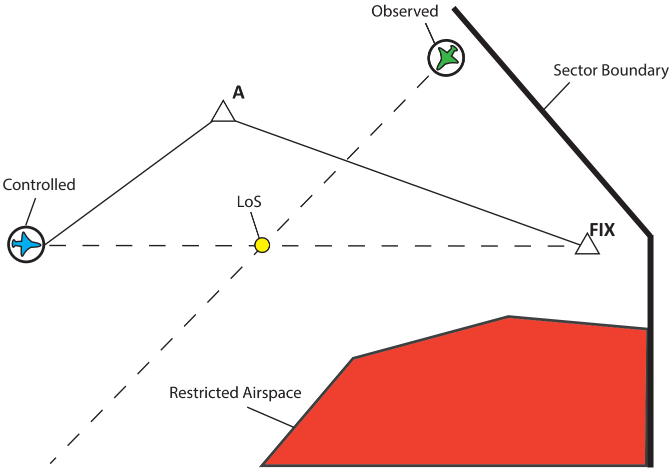


Fig. 2: Overview of the rerouting task. In order to prevent a LoS, the controlled aircraft is rerouted through point A to get to its target waypoint FIX.

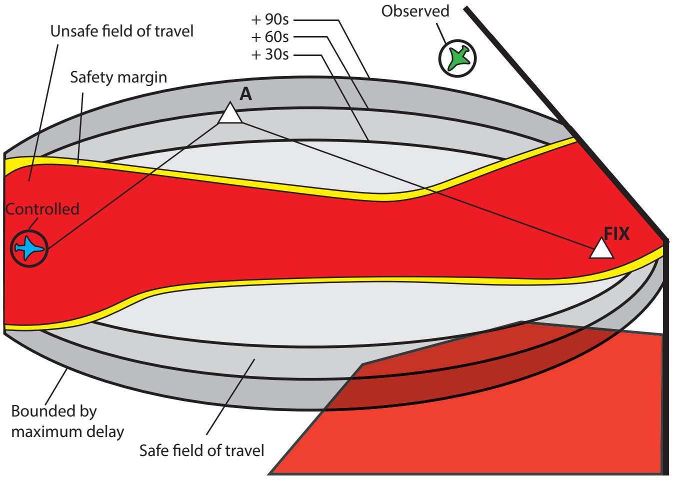


Fig. 3: Schematic overview of the TSR and how it can be used for the rerouting task.

outer ellipse is thus not limited by maximum velocity, but rather by a maximum allowable delay. Furthermore, maximum turning constraints have not been implemented in this version.

A schematic overview of the TSR has been provided in Figure 3. Note that this is the same traffic situation as in Figure 2. The TSR still represents the solution space for rerouting via directing the aircraft to an intermediate waypoint somewhere inside the TSR. Gray areas resemble the safe travel space, while the red area indicates the field of unsafe travel. The yellow area indicates a safety margin. Separation criteria are the conventional 5 NM with an addition of 1 NM for the safety margin. Finally, the gray ellipses indicate the optimality in terms of time delay. Lighter gray ellipses are more optimal than darker ones.

C. The Agent: The Flexibility Metric

The tasks of the Flexibility Metric algorithm [20]–[22] is to find an optimal conflict- free trajectory from the aircraft's current position to a target point inside the airspace sector. The additional flight time must not exceed the maximum allowable delay. The optimal solution depends on the strategy used by the algorithm. The Flexibility Metric has four important steps:

1) Defining a metric grid,
2) Finding all reachable cells: *forward propagation*
3) Finding all possible trajectories: *backward propagation*, and
4) Evaluate all feasible trajectories and choose the most optimal one

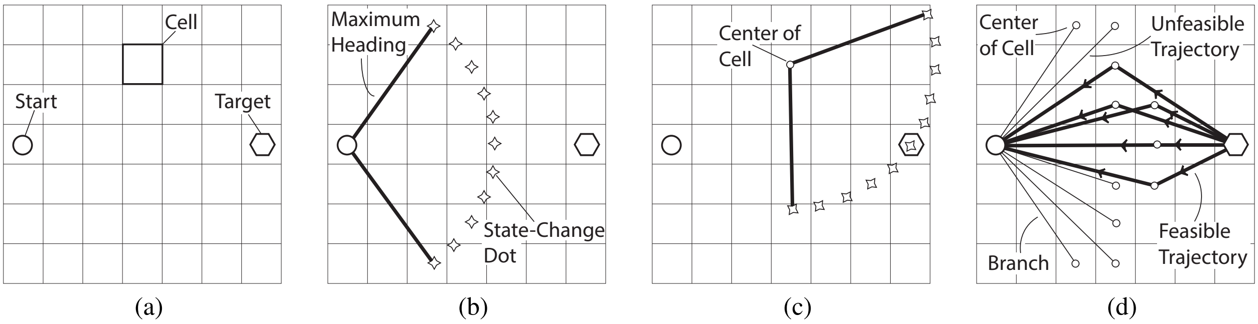
In Step 1, the algorithm discretizes the problem in space and time. Additional constraints are applied to the problem, which are related to the aircraft state-change capabilities and obstacles it needs to avoid. Step 2 uses the information from the metric grid to find reachable cells during forward propagation. From all those reachable cells, feasible trajectories are selected during backward propagation.

Figure 4 shows the forward and backward propagation steps of the algorithm. In Figure 4a, a top-down view of the rerouting problem is given. A square grid discretizes the problem in space. The algorithm will not look for a solution outside the sector. In Figure 4b, the algorithm starts with forward propagation. The algorithm searches for possible locations where the aircraft could be at a particular time interval. The locations are determined by the combination of state-change capabilities and time interval. The state-change capabilities in this research have a constant speed and are limited by a maximum heading change and resolution in which it samples potential intermediate waypoints for further evaluation. If a cell of the grid contains a state-change dot, the cell is under further evaluation. A branch is created from the start cell to the cell at the next time interval if it does not breach the separation criteria. In Figure 4c, this process is repeated once (one intermediate waypoint) starting from the centers of the cells of the next time interval.

During backward propagation, the algorithm investigates what the feasible trajectories are. As the name implies, this process starts at the target (end) waypoint. From there it follows all branches that are connected to it and sets their status to 'feasible'. In Figure 4d, the feasible branches are indicated by a bold line. Note that there are intermediate waypoints that are not set to feasible. This could be due to violating the separation criteria (no branch creation) or time constraint (the end-point is not reached within the maximum allowable time deviation).

In the last step, an optimal trajectories is chosen. This is done by means of a cost function that optimizes for one of the strategies. Two strategies have been implemented: shortest path and largest separation margin. During shortest path the algorithm is evaluating cells based on their trajectory length. In the separation margin strategy, the algorithm tries to find the trajectory that keeps the largest separation to the obstacles.

The Flexibility Metric was found to be a suitable match to the TSR [37] due to their similarity in strategy. Both algorithms are designed for aircraft rerouting by placing a single intermediate waypoint (dog-leg). They use the performance of the aircraft to investigate the reachable area. Subsequently,



(a) (b) (c) (d)

Fig. 4: Grid overview (a), forward propagation (b and c) and backward (d) propagation of the Flexibility Metric.

they exclude rerouting options that would result in conflicts. This results in a collection of feasible rerouting options from which, an optimal trajectory can be chosen that corresponds with the high level goals of the work domain. The main difference in approach is the discretized view the Flexibility Metric algorithm has of the work domain when compared to the TSR. If the settings of the algorithm are altered such that the size of the discrete cells is progressively reduced until they become indistinguishable from the continuous version, the Flexibility Metric becomes the TSR not considering decision selection. Due to the similarity, there is only a small gap in how information is integrated and therefore it is easier to combine the TSR with the newly developed interface elements.

Note that the algorithm has settings that could be altered. These settings affect the routing task of the algorithm. Insight into these settings plays an important role in creating more transparency. Therefore, they are used in the experiment to test the understanding of the participant with respect to the algorithm.

IV. INTERFACE DESIGN

The CWA [26] forms the basis of EID. One of the most insightful elements of this analysis is the work domain analysis. This analysis is used to select crucial information for visualization. This section discusses the work domain analysis. Furthermore, it is explained how the main findings of the work domain analysis are made visual. Finally, it is discussed how the interface supports human operators in their task of understanding automation.

A. Work Domain Analysis

The first step in the work domain analysis is making the system boundaries explicit. The work domain under consideration is an airspace, which is crossed by aircraft that need to travel through the airspace safely and efficiently. Normally, the work domain is independent of the actor that is working in it. However, as the goal is to make the inner-workings of the agent visible, it is chosen to incorporate the agent's view into the work domain. This creates insight into how elements of the agent are affecting routing.

A commonly used tool for mapping the work domain is the Abstraction Hierarchy (AH). The AH makes a connection between the physical basis and the purpose of the system using the relationships between five different levels of abstraction. At each level of abstraction, the level above and below are connected by a means-end relationship. The AH makes the relationships within the work domain salient and is therefore a basis for both the information content and structure [23].

The AH for this work domain can be found in Figure 5. In order to make a clear distinction between the physical view and the agent's view, two boxes were created that indicate the two object worlds as discussed by Naikar et al. [38]. The basis of the AH is retrieved from Van Paassen et al. [19].

1) Functional purpose

The main goal of the system is to ensure safety throughout the airspace. Aircraft need to be able to traverse the sector safely. If the safety goal is met, efficient travel is preferred. The system aims to produce safe and efficient travel for aircraft across the sector.

2) Abstract function

The underlying principles used to achieve the goals of the system are present at this level. The productivity is achieved by locomotion of aircraft through the sector. This is bound by both constraints related to the aircraft and external constraints such as the airspace sector. The main mean to create safety is via separation between aircraft, restricted airspace and dangerous weather. By maintaining a separation of at least 5 NM horizontally, safety standards are met. The driving factor for creating efficient trajectories is economy. Shorter routes are more economical and thus more efficient.

3) Generalized function

At this level the underlying processes involved can be found. Routing ensures locomotion of aircraft. It is also a means to ensure separation and economic flights. Routes need to be found that ensure enough separation from the obstruction in the airspace. This could be another aircraft, restricted airspace or dangerous weather. Locomotion is achieved by travelling through a volume and is also bounded by the performance capabilities of the aircraft.

4) Physical function

At the physical function, the components of the system are assembled. Note that a distinction is made between the physical view and the agent's view. In the physical view, the volume for travel is achieved by the airspace sector through which aircraft can traverse. Obstruction is created by air traffic and restricted airspace. Furthermore, routing is achieved by placing intermediate waypoints.

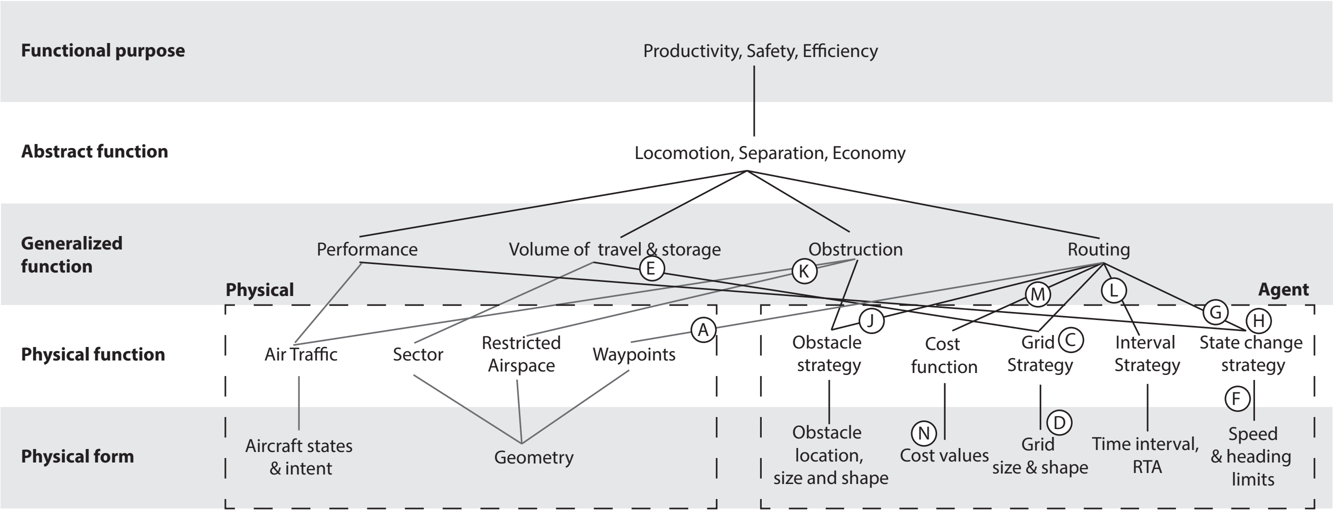


Fig. 5: Abstraction Hierarchy of the ATC system with an automated agent for rerouting. A distinction is made between the physical and agent's view on the system. Interface element IDs (A-N) are indicated in the AH.

How routing is affected can be found in the agent's view of the abstraction hierarchy. The agent is creating routes and is influenced by five elements. The first step in finding feasible trajectories is the creation of the grid. This limits rerouting by discretizing the space. Furthermore, routing is limited by a discretization of time by the interval strategy. The agent's view on the performance capabilities of the aircraft is stored in the state-change strategy. This has an important effect on the evaluation of possible intermediate waypoints and is thereby also affecting routing. The obstacle strategy stores information about obstructions. If trajectories violate the algorithm's separation criteria, these will not be further considered. The same holds for trajectories that exceed the maximum allowable delay as defined in the interval strategy. Both strategies are thus affecting routing. The final element that is crucial for generating trajectories is the cost function as it is used to select an optimal trajectory.

5) *Physical form*

At this level, the states of the components are described. For the sector, restricted airspace and waypoints, the physical form is primarily geometry. For air traffic, these are the aircraft states and intent.

In the agent's view, the physical forms are as follows. For the obstacle strategy, these are the obstacle locations, size and shape. The physical form of the cost function are the cost values. The grid strategy has the grid itself including its size and shape. The physical form of the interval strategy are the particular time intervals at which the algorithm tries to find a solution as well as the required time of arrival (RTA). This includes the maximum allowable delay. Finally, the state-change strategy has the limitation in heading and speed at this level.

B. Structure and Functionality

During the development of the interface, the goal was to visualize the elements and means-end links of the AH. Note that the interface element IDs are also present in the AH of Figure 5. In order to make the agent more transparent, the interface elements were focused on the agent's part of the AH and its connection to routing. By visualizing these elements and links, a greater insight is provided into how the algorithm is operating and what constrains are driving its decisions.

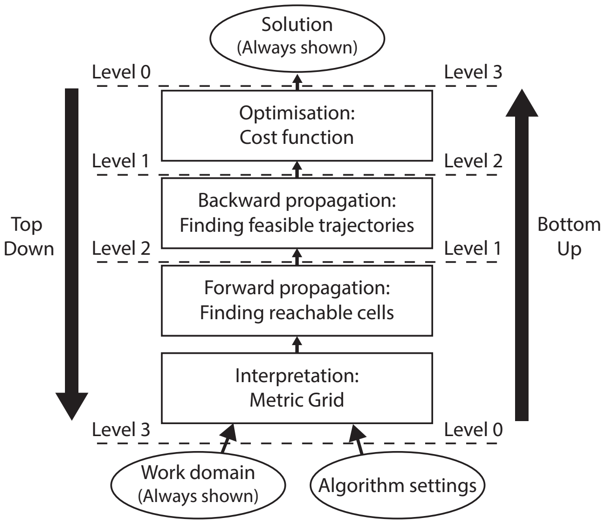


Fig. 6: Top-down and bottom-up approach with respect to the steps of the Flexibility Metric algorithm.

In order to facilitate both top-down and bottom-up processing, transparency has been implemented in two different directions. Figure 6 gives an overview of the algorithm's high level steps and indicates how the levels of transparency are related to that. In both directions the domain (TSR) and agent's solution are always shown as the focus is on the inner-workings of the agent. The levels in the bottom-up approach were chosen to match the last three steps of the algorithm. After defining the metric grid, the cells under consideration are shown. Then these cells are checked if they result in feasible trajectories. Finally, an optimal trajectory is chosen.

In the top-down approach, the opposite is the case. Starting from the solution rationale, it is investigated what the other potential trajectories are and why these are not chosen. Then, blocked and out-of-time trajectories are shown. Finally, the constraints stored in the metric grid are visualized. In Figure 7, the resulting interfaces tied to these levels are presented.

Figure 7a shows the interface with no additional support. The controlled aircraft ① has target waypoint EXOBA ②. It needs to deviate from the direct route, because it is in conflict with the observed aircraft ③. The new route that is proposed by the automation is the cyan line Ⓐ. Furthermore, the sector boundary ④, a restricted airspace ⑤ and waypoints ⑥ are indicated on the screen. Figure 7b shows Level 0 of both the bottom-up and top-down approach. At this level, the TSR Ⓑ can be seen, which forms the basis for the other interface elements. These elements are grouped in three levels. The levels are first explained in the bottom-up approach and then the differences of the top-down approach are indicated.

1) Bottom-up Level 1

At this level (Figure 7d), constraints from the metric grid driving the algorithm's reasoning process are shown. This information is used to show which cells will be further evaluated. The grid Ⓒ and cell sizes Ⓓ that automation uses are shown. Because the algorithm does not consider the grid outside the sector, the remaining shape, called the evaluated area boundary Ⓔ, is highlighted.

The maximum heading change is visualized by the blue lines Ⓕ. Only the area in between the two blue lines, is still under consideration. The state-change bands Ⓖ show all positions at which the aircraft could be at a particular time interval. In the example of Figure 7, the algorithm is evaluating three different time intervals and therefore three state-change bands are drawn. The state-change band is sampled with state-change dots Ⓗ at particular heading increments. In this example, heading increments of 5 degrees are used. If a grid cell contains such a state-change dot, the cell will be under further evaluation and is called an evaluated cell Ⓘ.

2) Bottom-up Level 2

At this level (Figure 7f), cells that do not match the separation criteria or maximum allowable delay are colored red Ⓙ and blue Ⓛ respectively. These cells will not be under further evaluation. Interaction at this level is possible. By hovering with the mouse over the cells Ⓝ, the status of that cell is presented. This could be *Blocked* for cells that do not meet the separation criteria. The status *Time-out* is shown for cells that do not meet the time constraint. All evaluated cells that remain are considered feasible and are assigned the status of *Feasible*. All cells that are not under evaluation get the status of *Unevaluated*. For blocked cells, the reason Ⓚ for that cell to be blocked is highlighted as well. This can be an aircraft or restricted airspace.

3) Bottom-up Level 3

The optimality of feasible cells Ⓜ is presented in Level 3 (Figure 7g). This is shown by different shades of green. Darker green cells are considered less optimal, while lighter cells are more optimal. Additional information about the relative cost percentage Ⓝ is portrayed while hovering over feasible cells at this level. The percentage describes the difference in costs between the feasible and most optimal cell. Furthermore, the extra distance of the trajectory to the cell, compared to the direct path, is shown. Finally, the shortest distance to an obstacle at any point of the trajectory, is provided.

4) Top-down levels

Level 0 and Level 3 of the top-down approach are equal to the bottom-up approach. The intermediate levels are different. In Level 1 (Figure 7c) of the top-down approach, the feasible cells Ⓜ are drawn directly. Also the extensive hover information Ⓝ is already provided at this level. Level 2 (Figure 7e) adds the blocked cells Ⓙ and time-out cells Ⓛ as extra information. Hovering over these cells still shows the status of that cell Ⓝ. Only at Level 3, the other constraints that drive the algorithm's reasoning process are shown.

C. Relationship to Transparency Constructs

Note that the concept of counterfactual explanations is used in this interface in a visual form. The constraints of the agent show the conditions for which the current cells are evaluated. It can be imagined how changes in, for example, the time intervals or maximum heading lines could lead to the evaluation of other cells instead of the current ones.

Furthermore, "why-not" explanations are encouraged by the possibility of getting information while hovering over the cells. In this manner, it is easier to explore the field of possibilities and get direct feedback on why certain cells are chosen and why others are not.

Finally, showing extra information by hovering with a mouse cursor is in accordance with the concept of progressive disclosure. Extra information is only given when the user is actively requesting it by hovering over the cells.

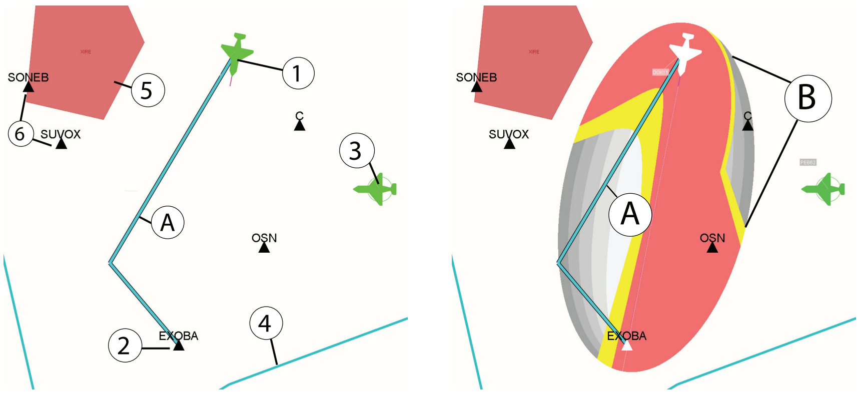
D. Behavioral Support

The skills, rules and knowledge (SRK) taxonomy [32] is a framework for describing the control behavior of the human operator. Three levels of cognitive control behavior are described in the taxonomy: skill-based behavior (SBB), rule-based behavior (RBB) and knowledge based behavior (KBB). In general, lower levels of control are faster, require less cognitive effort and are less error-prone [23]. Even though KBB takes more effort, it is required in unfamiliar events. Interfaces should therefore support all levels of control behavior.

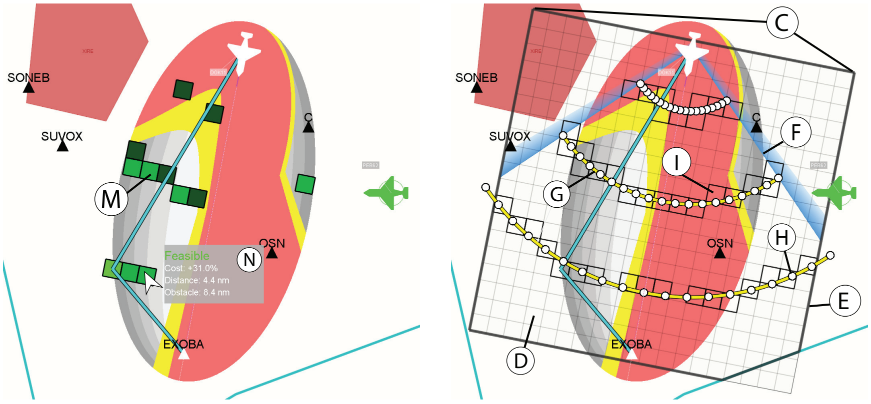
The task of understanding automation is mainly knowledge driven for novice users. The interface makes the knowledge driven task more efficient by creating perceptional elements that support mainly RBB. SBB is only triggered in minimal amounts. Important to note is that training can reduce the level of control behavior required for the task. How the interface supports these levels of cognitive control is described below.

1) SBB

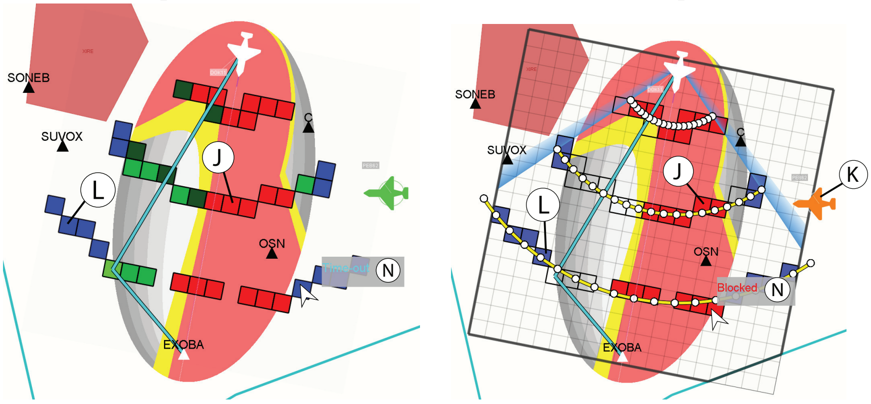
The elements that trigger SBB are those implemented with interaction. Retrieving extra information by hovering over cells can support SBB. Furthermore, hovering over blocked cells highlights the reason for that cell to be blocked. This immediately makes a connection between the cell and the obstacle, which is a means-end link in the AH.



(a) No Support (b) Bottom-up & Top-down Level 0



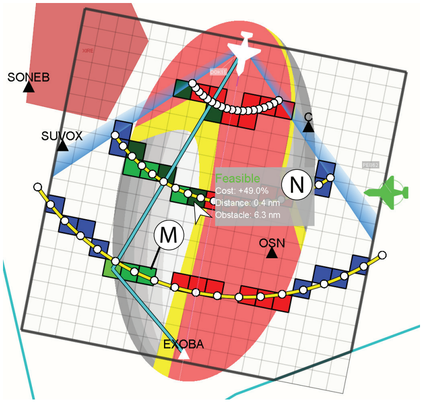
(c) Top-down Level 1 (d) Bottom-up Level 1



(e) Top-down Level 2 (f) Bottom-up Level 2

Legend

ID	Elements	ID	Elements
Ⓐ	Proposed trajectory	Ⓗ	State-change dots
Ⓑ	TSR	Ⓘ	Evaluated cells
Ⓒ	Grid	Ⓙ	Blocked cells
Ⓓ	Cell size	Ⓚ	Blockage reason
Ⓔ	Evaluated area boundary	Ⓛ	Time-out cells
Ⓕ	Maximum heading lines	Ⓜ	Feasible cells
Ⓖ	State-change bands	Ⓝ	Hovering information



(g) Bottom-up & Top-down Level 3

Fig. 7: Interface overview for all levels of the bottom-up and top-down approach.

2) RBB

RBB is supported by signs that trigger a set of rules known by the operator. Multiple elements of the AH have been implemented as perceptual clues supporting RBB. Color coding of cells has a direct relation to the status of the cell (e.g., red color resembles blocked cells). Furthermore, green color shading triggers an idea of the optimality of cells and thereby the strategy that is used by automation.

Also, the constraints presented in the interface such as the grid, heading lines and state-change bands are used to trigger rules. For example, it can be immediately seen that points that fall outside the grid are not evaluated. The same holds for cells that lie behind the heading lines or just outside of the state-change bands. These elements represent rules that show which cells are under evaluation and which cells are not. This creates an understanding of the decision process and thus information supporting RBB helps in achieving understanding of SAT Level 2.

3) KBB

Symbols encourage KBB. The additional information presented to the operator while hovering over cells, can support KBB. The values for cost, distance and separation can be used to elicit what strategy is used by automation.

Furthermore, the constraints such as the grid, heading lines and state-change bands resemble a model of automation which supports reasoning about other (hypothetical) options. For example, the visualization of the arc-shaped state change band and state change dots allow the operator to make a prediction about the algorithm's settings of the heading resolution and time interval. This knowledge can then be used to reason whether these settings need to be changed in order to create "better" solutions. In this manner, KBB may lead to understanding why other cells are not feasible, blocked, evaluated, etc and what settings should be altered in order to change this. By supporting KBB, the interface allows the operator to make a prediction about the algorithm and therefore contributes to developing understanding via SAT Level 3.

V. EXPERIMENT DESIGN

In order to evaluate the interface, a human-in-the-loop experiment was performed. During the experiment, the main goal is to find whether the interface is able to provide actual and perceived understanding. Furthermore, a core research goal is to find if a top-down or bottom-up method is preferred. The experiment design is described here.

A. Participants

In total, ten participants took part in the experiment. Participants' ages ranged from 23 to 47 years (M = 27.4, SD = 7.5). They were either students or staff of the Control and Operation department of the Aerospace Engineering Faculty of the TU Delft. All participants followed courses on human-machine systems and ATC.

The scope of the experiment is to define what is required for creating understanding of automation through transparency rather than investigating an operation-ready application. The above mentioned group of participants fits this scope, because they have general knowledge of the ATC system, but are not biased by prior knowledge or by experience as real ATCOs. Furthermore, ATCOs are known to be critical towards adopting automation that aims to (partly) take over their job, which might affect results as well.

B. Apparatus

The experiment was performed in the SectorX simulation software developed at the Control and Simulation section of the Faculty of Aerospace Engineering at the TU Delft. The simulation was shown on a 30-inch monitor with a resolution of 2560 x 1600 pixels. Participants could only give input using the mouse. The screen was duplicated on a separate monitor such that the investigator could observe the participant's behavior.

C. Experiment Tasks

During the experiment, the participants were given a static scenario in which two aircraft were in conflict and rerouting is required. Automation proposed a new route to solve the conflict. The task of the participant was to answer questions about the automation's decision. These questions, four in total, needed to be answered at all levels of transparency; starting at Level 0. They had to answer these questions to the best of their ability, but within a time limit of 7.5 minutes per scenario. Figure 8 shows the display presented to the participant.

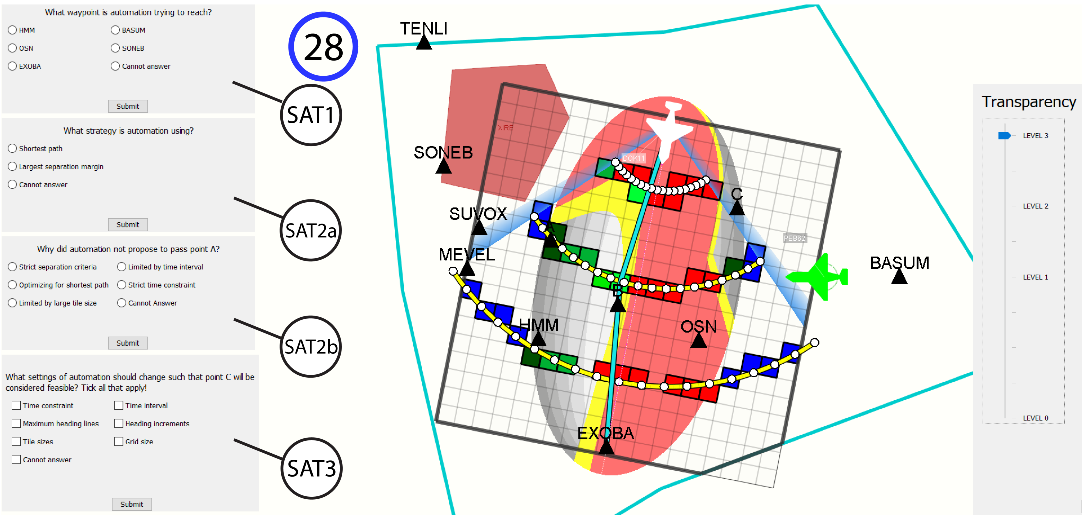
All questions were multiple choice and cover the 3 levels of SAT. The first three questions allowed selection of only one answer option, while for the last question multiple options could be selected. All questions contained a "Cannot answer" option that was encouraged to be used if they felt they had insufficient information to properly answer the question. After submitting a question, the participant was asked to indicate their confidence using the dialog shown in Figure 9. Once all question were answered, the level of transparency increased and they could then either adjust their answers, or submit the same answer. This process continued until all question were answered at the highest level of transparency. Once this was done, the next scenario started.

D. Independent Variables

The experiment had three independent variables. The first one is the level of transparency and is varied within-participants. The research aim is to find out if higher levels of transparency of this interface actually increase understanding and therefore all participants had to progress through each level of transparency.

The second independent variable is the order of transparency level and is varied within and between participants. The orders that are investigated are the top-down and bottom-up approach. All participants encountered both approaches. Group A started with bottom-up and Group B started with top-down.

The third within-participant independent variable is the automation setting. This variable has been chosen in order to generate a large variety of relevant scenarios that affect routing of automation. Thereby, a larger understanding of the algorithm can be tested which is not limited to a specific setting of automation.



What waypoint is automation trying to reach?

○ HMM ○ BASUM
○ OSN ○ SONEB
○ EXOBA ○ Cannot answer

What strategy is automation using?

○ Shortest path
○ Largest separation margin
○ Cannot answer

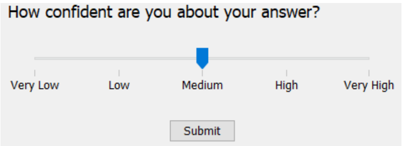
Why did automation not propose to pass point A?

○ Strict separation criteria ○ Limited by time interval
○ Optimizing for shortest path ○ Strict time constraint
○ Limited by large tile size ○ Cannot Answer

What settings of automation should change such that point C will be considered feasible? Tick all that apply!

☐ Time constraint ☐ Time interval
☐ Maximum heading lines ☐ Heading increments
☐ Tile sizes ☐ Grid size
☐ Cannot answer

Fig. 8: Overview of display presented to the participant. On the left of the screen, the questions about automation are shown (SAT 1 through SAT 3). At the top left corner, a clock shows the remaining time. In the middle, the air traffic situation is displayed and on the right the current level of transparency is indicated.



How confident are you about your answer?

Fig. 9: Confidence question dialog which appears after submitting a question.

TABLE I: Overview of scenarios used in the experiment.

ID	Setting changed	Strategy	Aircraft	Restricted airspace
CEL	Cell size	SP	2	Yes
INT	Time interval	LSM	3	No
SEP	Separation criteria	SP	2	No
TIM	Time constraint	LSM	2	No
STR	Strategy	SP	2	Yes

E. Scenarios

During the scenario design, the goal was to develop scenarios of equal difficulty that focused on different parts of the algorithm's reasoning process such that a more complete image of the participant's understanding of automation can be tested. Five settings were chosen that have an influence on routing via the grid strategy, state change strategy, interval strategy, cost strategy and obstacle strategy as defined in the AH of Figure 5. Table I shows an overview of all scenarios. Scenario CEL changed the cell size, which effected the discretization in space by creating larger cells. In Scenario INT, the amount of time intervals changed which affected the state-change bands locations. The separation criteria of the agent was changed in Scenario SEP resulting in the algorithm being more strict in determining conflicts. The maximum allowable delay of the algorithm was changed in Scenario TIM making the algorithm more strict in determining whether a solution met the time constraint. Finally, in Scenario STR the strategy was the dominant reason for making the solution non-trivial.

The sector that is used in the scenarios is the MUNSTER sector of the Maastricht Upper Area Control. Scenarios contained two or three aircraft and could be combined with a restricted airspace. In each scenario, an aircraft is selected that needs to be rerouted in order to solve a conflict. The strategy, either shortest path (SP) or largest separation margin (LSM), is alternated between scenarios. The five scenarios were tested in the top-down and bottom-up approach. In order to prevent recognition, the scenarios of the top-down approach were rotated 90 degrees.

F. Control Variables

A number of variables is controlled such that the influence of these variables is isolated. During the experiment, scenarios are static, because because traffic situation change over time and therefore the algorithm might suggest another trajectory. Furthermore, time was limited to a maximum time per transparency level in order to prevent that participants take all the time they desire. Time limits were not too strict as this could discourage participants to fully understand the situation. Participants had 3 minutes for answering the questions at Level 0 and 1.5 minutes for the other three resulting in a maximum time of 7.5 minutes for the complete scenario. In Level 0 more time was allowed, because participants had to read the questions at this level. These time limits were found during pre-testing.

Furthermore, speed and altitude are kept constant during the experiment. En-route ATCOs almost never use speed to solve conflicts. Furthermore, keeping these variables constant made rerouting via an intermediate waypoint the only option to solve the conflict.

G. Dependent measures

The first dependent measure to be measured is understanding. During the experiment, both actual and perceived

TABLE II: Overview of questions asked in a scenario.

ID	SAT Level	Question
Q1a	SAT 1	What waypoint is automation trying to reach?
Q1b	SAT 1	How did automation propose to fly?
Q2	SAT 2a	What strategy is automation using?
Q3	SAT 2b	Why did automation not consider Point A?
Q4	SAT 3	What settings should change to make Point B feasible?

understanding are measured. Actual understanding is measured by queries that correspond with the three levels of SAT. The questions can be found in Table II and are equal for all scenarios and levels. The first question is about the goal and plan of automation. Two versions, Q1a and Q1b, are varied between scenarios. The second question is about the strategy that the algorithm is using, which relates to the second level of SAT. The third question always asked why another intermediate waypoint was not chosen. These correspond with the altered setting of automation. This question is considered as an additional measure of the second level of SAT. The last question is about making predictions of the algorithm, which is SAT Level 3, by asking the participant what settings of the algorithm should change in order to make another cell considered feasible. In this question, other settings of the algorithm, apart from the changed setting, are taken into account such as the maximum heading lines, heading resolution and grid size. Note that multiple answers can be selected for the last questions. All questions have the "Cannot answer" option in case the participants believe that they do not have enough information to answer the question.

In addition to measuring their actual understanding, their perceived understanding is measured as well. Once a question is submitted, participants indicate their confidence level on a scale of 0 to 100 using the dialog of Figure 9. Furthermore, the time to answer questions and the number of interactions that the participants make by hovering over cells, are measured. Only interactions longer than 0.5 seconds were taken into account in order to filter out insignificant interactions.

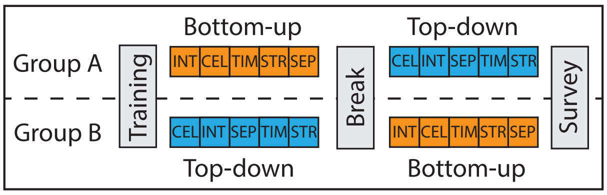


Fig. 10: Schematic overview of the procedure during the experiment. Scenario order is mixed using latin square. Scenarios are abbreviated by their ID which are stated in Table I.

H. Procedures

An overview of the procedure can be found in Figure 10. The participant received a briefing two days in advance containing basic info about the experiment, TSR and rerouting task. Before the experiment, the participant was trained to use the interface. Four scenarios were used to explain the interface elements, questions they needed to answer and time clock. Subsequently, all scenario types were treated in five scenarios. In total, nine scenarios were divided over two transparency directions. The scenarios for explaining the questions and time-clock were very similar and therefore bottom-up had an extra scenario compared to top-down.

After the training phase, the participants were divided into two groups. Group A started with five scenarios in the bottom-up order and then five scenarios in the top-down order leaving a break in between both sessions to prevent recognition of scenarios. Group B started with the top-down approach and then switched to bottom-up. The scenarios were shuffled using latin square in order to balance all experiment conditions.

The experiment thus has a mixed design of within- and between-subjects, which allows for measuring the difference between the top-down and bottom-up approach while being able to compensate for the order in which these transparency orders are presented.

Once both sessions were completed, the participants were asked to fill out a post-experiment questionnaire in which they could give feedback on the interface and their preference regarding interface elements and transparency order.

I. Hypotheses

The experiment is guided by two main hypotheses. The first hypothesis is that the interface is able to create a larger understanding due to higher levels of transparency. The idea is that more insight into the agent also provides a better understanding. Both actual and perceived understanding are measured in order to test this hypothesis.

The second hypothesis is that the top-down approach is preferred over the bottom-up approach. As indicated by Vicente et al. [23], top-down processing is more efficient. This hypothesis is tested by measuring the actual and perceived understanding. Additionally, the time to complete the task is measured. If the top-down approach is more efficient, it is expected that scenarios with the top-down approach take less time to fulfill. Furthermore, the number of interactions are measured. Higher levels of interaction can indicate a more active participation.

J. Data Processing

Data of all ten participants were used for statistical analysis. Data did not fulfill the homogeneity of the variation and/or normality assumptions and therefore all data were tested using non-parametric tests. For related samples, the Friedman test was used. In order to find pairwise comparisons, a Wilcoxon test was performed. The Bonferonni correction was used to compensate for multiple tests. The Mann-Whitney U test was used to find significances between Group A and B. A significance level of 0.05 was used for all tests.

VI. RESULTS

The result section is divided into four parts. The first three sections describe the results of the three dependent measures: understanding, time and interaction. The last section reports results that were found in the post-experiment questionnaire. Scenarios are indicated with the symbols shown in Table I.

A. *Understanding*

The task of the participant in the experiment was to answer questions about automation at four different levels of transparency. All participants answered SAT 1 correct for all scenarios at all levels. Therefore, only results for SAT 2 and 3 are further discussed.

The raw data of one participant for SAT 2a are shown in Figure 11. The figure shows how the participant answered SAT 2a for all scenarios for all levels of transparency. Note that at lower levels of transparency participants either answered "Cannot answer", answered falsely or answered correctly with low confidence. At higher levels of transparency, this participant answered the question correctly with high confidence in all scenarios.

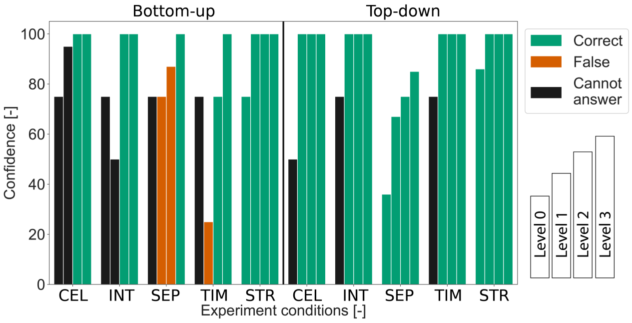


Fig. 11: Raw answer data of SAT 2a for one participant.

In order to make statistical claims and better visualize trends with respect to all participants, the data were further processed. A hit ratio is calculated, which shows the percentage of correctly answered questions with respect to the number of given answers. It aims to give insight into the actual understanding of the agent's decision.

Hit ratio is calculated per participant, per question, per level, per direction. Therefore, the percentage is calculated over five answers as five scenarios were done per participants per direction. The choice to average over scenarios was justified by the aim to make the scenarios of equal difficulty. Furthermore, averaging over all these scenarios results in a hit ratio that takes into account a variety of aspects of the agent.

"Cannot answers" were not counted as correct for this percentage. Although these type of answers can show insight into the overall understanding of the agent. They do not resemble the actual understanding of the agent's decision, which is what the hit ratio aims to show.

This choice triggered a problem for the interpretation of confidence. It is clear that participants who had a high confidence of not being able to answer should not be counted the same as participants who falsely answered with very high confidence. Therefore, the confidence was inverted such that a more logical meaning is given to this answer option. This results in counting high confidence "cannot answers" as a "not correct" answer with very low confidence. The confidence is then also averaged over five scenarios.

To get more insight into how both hit ratio (HR) and confidence (CON) develop over the level of transparency, both measures for understanding were plotted with respect to each other in Figure 12. There is a plot for each question and transparency direction. Note that each dot resembles an average of one participant at one level. Confidence intervals of 95% are drawn per level. Black lines connect the center points of these ellipses. The plots are analyzed per question.

1) *SAT 2a*

The results of SAT 2a can be found in Figure 12a and 12d. In both figures, participants had higher hit ratios and confidence ratings at higher levels of transparency. Furthermore, a difference can be seen between the two transparency directions. The bottom-up approach has a more gradual increase of both confidence and hit ratio, while in the top-down direction participants were able to answer correctly already at Level 1 in almost all cases. This corresponds with the results of the post-experiment survey in which participants indicated to have a preference for answering SAT 2a using the top-down approach.

There was a significant effect found between transparency level and hit ratio ($\chi^2(3) = 27.702, p < 0.01$) and between transparency level and confidence ($\chi^2(3) = 30.000, p < 0.01$). Post-hoc tests indicated that the difference between Level 0 and Level 2 was significant (HR: $p < 0.01$; CON: $p = 0.003$) as well as between Level 1 and Level 3 (HR: $p = 0.034$; CON: $p = 0.003$). Top-down outperformed the bottom-up approach. Wilcoxon test showed a significant effect between bottom-up and top-down at Level 1 (HR: $Z = -2.877, p = 0.016$; CON: $Z = -2.803, p = 0.020$). At Level 2 only confidence was found to have a significant effect (HR: $Z = -1.857, p = 0.252$; CON: $Z = -2.805, p = 0.020$). At the other levels, no significant effect between top-down and bottom-up was found.

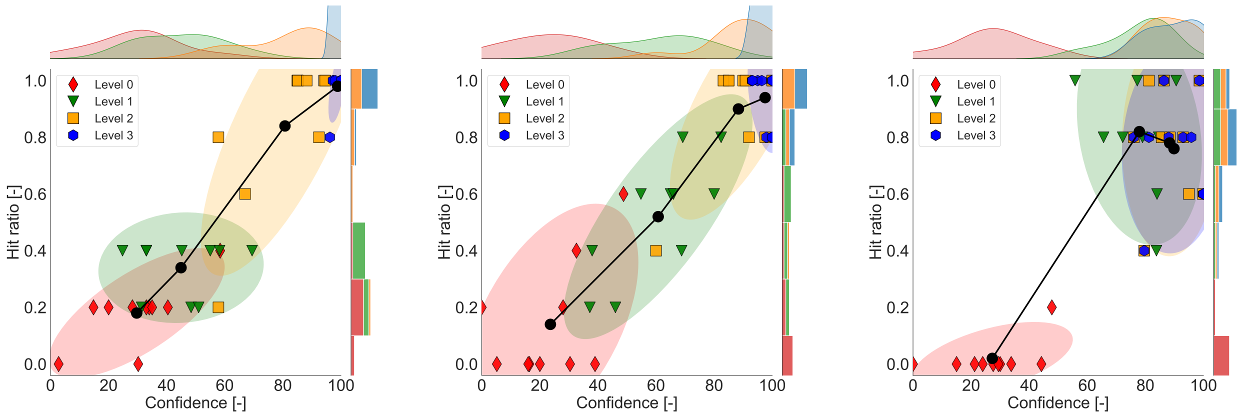
2) *SAT 2b*

Figure 12b and 12e show the bottom-up and top-down approach of SAT 2b. Note that both plots still show that increasing transparency yields higher hit ratios and confidence levels. However, there is no clear distinction between the two directions. The two lines connecting the centers of the ellipses follow more or less the same path.

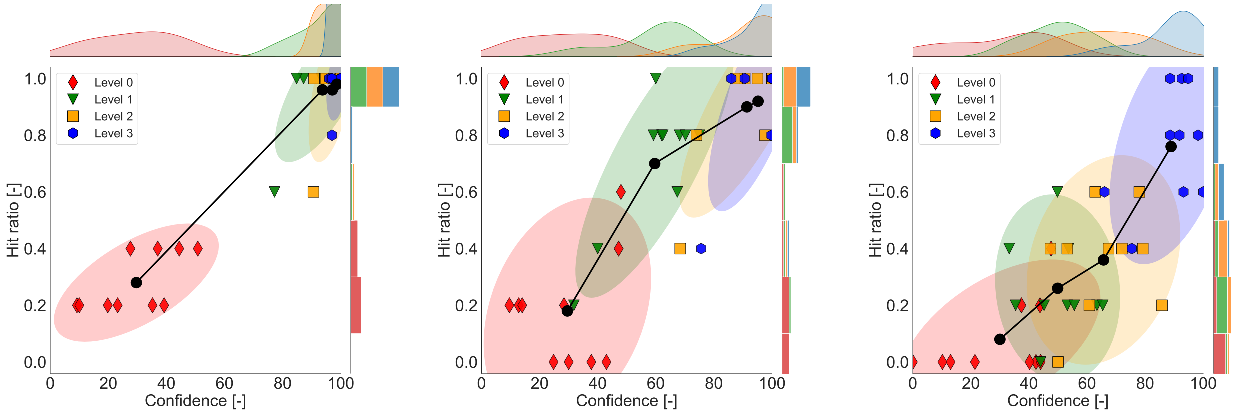
Also at SAT 2b, a significant effect was found between transparency level and hit ratio ($\chi^2(3) = 29.478, p < 0.01$) and between transparency level confidence ($\chi^2(3) = 29.510, p < 0.01$). Post-hoc tests indicated that the difference between Level 0 and Level 2 was significant (HR: $p < 0.01$; CON: $p = 0.002$) as well as between Level 1 and Level 3 (HR: $p = 0.034$; CON: $p = 0.006$). Wilcoxon test between top-down and bottom-up approach showed no significant effect between transparency directions at any levels.

3) *SAT 3*

In Figure 12c and 12f, the results for SAT 3 can be found. The bottom-up plot shows participants were able to correctly answer questions at lower levels of transparency, while the top-down plot is more scattered showing a much more gradual increase of confidence and hit ratio. This corresponds with the post-experiment questionnaire, in which participants indicated to prefer the bottom-up approach for SAT 3. Participants pointed out that this was the most difficult question and sometimes ambiguous. Note that the confidence ellipse of

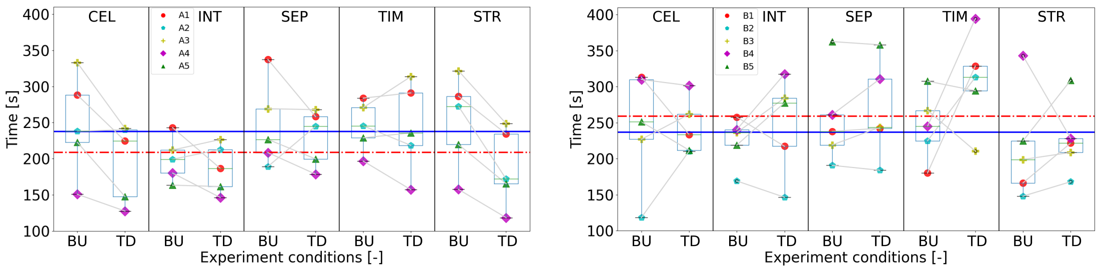


(a) SAT 2a: Bottom-up (b) SAT 2b: Bottom-up (c) SAT 3 :Bottom-up



(d) SAT 2a: Top-down (e) SAT 2b: Top-down (f) SAT 3: Top-down

Fig. 12: Hit ratio and confidence level plotted for SAT 2a, SAT 2b and SAT 3 at both transparency directions. Confidence ellipses of 95% are drawn per level and their center points are connected with a line.



(a) Group A (b) Group B

Fig. 13: Time to complete scenarios. Links connect data points of the same participant. Blue lines and red dotted lines show the means of the bottom-up (BU) and top-down (TD) approaches, respectively.

Level 3 in SAT 3 is larger compared to other questions, which indicates a larger variation in hit ratio and confidence.

In addition to SAT 2a and 2b, SAT 3 also indicates that a significant effect is measured between transparency level and hit ratio ($\chi^2(3) = 29.266, p < 0.01$) and between transparency level and confidence ($\chi^2(3) = 30.000, p < 0.01$). Post-hoc tests indicated that the difference between Level 0 and Level 2 was significant (HR: $p = 0.002$; CON: $p = 0.003$). The difference between Level 1 and Level 3 was only significant for confidence ($p = 0.003$). In SAT 3, the bottom-up direction outperformed top-down. Wilcoxon test showed a significant effect between bottom-up and top-down at Level 1 (HR: $Z = -2.829, p = 0.020$; CON: $Z = -2.805, p = 0.020$) and at Level 2 (HR: $Z = -2.751, p = 0.024$; CON:$Z = -2.803, p = 0.020$).

It can be concluded that higher levels of transparency are positively correlated with higher hit ratios and higher confidence levels. Furthermore, top-down yielded better results in SAT 2a while bottom-up performed better at SAT 3. In SAT 2b no significant differences were found.

This corresponds with the post-experiment questionnaire, in which participants indicated that higher levels of transparency made them better equipped to answer questions and understanding the agent when compared to lower levels of transparency. Furthermore, participants were divided in their preference for the transparency directions, which corresponds with these results as performance of transparency directions depended on the questions asked.

B. Time

The order in which questions are answered is not discussed as almost all question were answered from top to bottom. Figure 13 shows how much time the participants took to complete the scenarios. Statistical analysis showed that there was a significant difference between scenarios ($\chi^2(4) = 12.978, p = 0.003$). Pairwise comparisons showed a significant effect between the time interval (INT) and time constraint (TIM) scenarios ($p = 0.004$) and the strategy (STR) and time constraint (TIM) scenarios ($p = 0.030$). No significant effects were found in other pairs.

Wilcoxon test showed no significant effect between directions ($Z = -0.255, p = 0.799$). However, a significant effect within Group A ($Z = -2.023, p = 0.043$) and Group B ($Z = 2.023, p = 0.043$) was found between the top-down and bottom-up approach. In Figure 13, note that in Group A, the top-down approach requires significantly less time than the bottom-up approach. The opposite is true for Group B in which the top-down approach requires more time than bottom-up. These effects could be explained by learning effects that still occurred in the experiment.

Even though no significant effects were found between groups in the bottom-up ($U = 12.000, p = 1.000$) and top-down approach ($U = 20.000, p = 0.151$), it is remarkable that the mean times of groups in top-down have a difference of 50.32 seconds. Top-down approaches require prior-knowledge for effective usage. Insufficient training and a lack of prior-knowledge seem like plausible reasons for the longer time required for the top-down approach at Group B. More participants are needed to check if this effect significantly occurs.

C. Interaction

In order to filter out insignificant interaction, only hover events over cells longer than 0.5 seconds were taken into account. Changing the threshold to 0.1 or 1.0 seconds resulted in similar distributions. Figure 14 shows the interactions of the participants in which each data points resembles the total number of interactions in one scenario. In Group B participants had more interactions using the top-down approach compared to bottom-up. This difference is not clear in Group A. As Group B had a low amount of prior knowledge in the top-down approach, they might have been more inclined to use the interaction feature to retrieve extra information. In Group A, this effect does not appear, possibly because participants already have more prior-knowledge of the agent.

A significant difference was found between scenarios ($\chi^2(4) = 12.978, p = 0.011$) and pairwise comparison showed only a significant difference between the time interval (INT) and time constraint (TIM) scenario ($p = 0.011$). No significant effect was found between transparency directions ($Z = 1.686, p = 0.092$). Furthermore, a Mann-Whitney U test did not find any significant differences between Group A and B in both bottom-up ($Z = 0.602, p = 0.690$) and top-down approach ($Z = 0.104, p = 1.000$). Nonetheless, within Group B a significant effect was found between bottom-up and top-down ($Z = 2.023, p = 0.043$). No significant difference was found within Group A ($Z = -0.962, p = 0.336$).

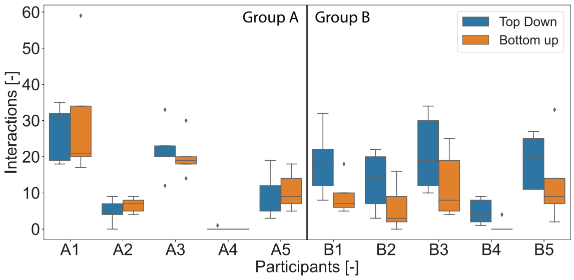


Fig. 14: Interaction for participants of Group A and B. A distinction is made between the two transparency directions.

D. Interface Preferences

Figure 15 shows an overview of the preferred transparency direction for different situations. In the experiment, an equal number of participants were in favor of bottom-up and of top-down. For ATC training, the majority of participants indicated that they were neutral on the preferred method. Some participants indicated that both transparency directions could be useful during training. The bottom-up method for creating an augmented insight into the workings of the algorithm and the top-down as a more pleasant direction in terms of how information is provided. During ATC operation, the majority of participants indicated that the top-down method would be most suitable. They stated that the most important decision-making information is shown earlier in this direction. Furthermore, gradual increase of information was preferred by participants in order to prevent cluttering and distractions.

Participants were asked to rate all interface elements in terms of usefulness. These ratings can be found in Figure 16. The *proposed trajectory* and the *feasible cell* were found to be especially valuable. Generalization of these ratings should be made with care as these ratings are very dependent on the task at hand. For example, the *highlighting of blockage reason* was rated very low, because people did not need to use that element for any question in the experiment. Another important sidenote is that the *evaluated area boundary* and *grid* both scored relatively low. Participants used only one of the two and subsequently gave the other element a lower score.

VII. DISCUSSION

The first hypothesis of the experiment stated that higher levels of transparency could increase understanding of automation. Results show that for all SAT levels, except SAT 1, the hit ratio and confidence level significantly increased for higher levels of transparency. Therefore, the interfaces

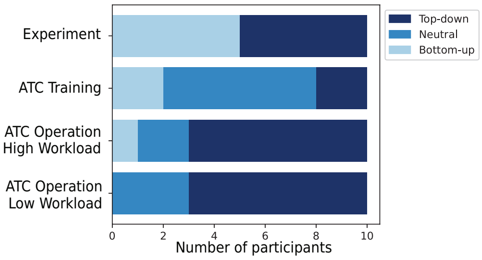


Fig. 15: Preferred transparency direction in four situations.

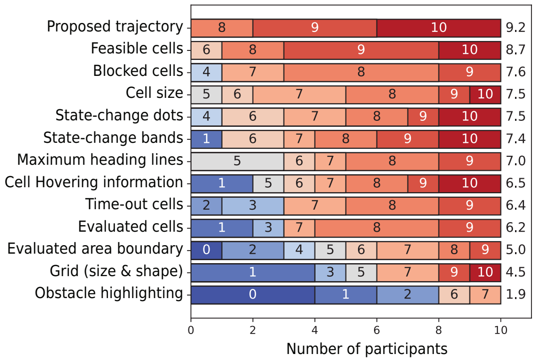


Fig. 16: Rating of interface elements on a scale of 0-10 for usefulness. Score is noted in the bars. Average score is displayed to the right of the bar.

increased both the actual and perceived understanding of SAT 2 and 3. This indicates as well that people were able to predict their own performance, which is favorable as it is not desired that people are over- or under-confident. However, it should be noted that participants had more difficulty with SAT Level 3. Supporting what-if probing of automation settings could enhance prediction of the agent.

The second hypothesis stated that the top-down approach would yield better results and would be more preferred by participants. The results of the experiment and post-experiment questionnaire do not directly indicate which direction was better. In terms of understanding, the top-down approach worked better with respect to questions related to the strategy of automation. However, if predictions about the agent needed to be made, the bottom-up approach yielded better understanding.

Significant results within groups were found in between transparency directions both in time and interaction. These findings show that participants still learned during the experiment. Therefore, the above mentioned results might not be valid for experienced ATCOs. More extensive training is advised for future research. Furthermore, the ability of making predictions about the algorithm, as measured in question SAT 3, might be interesting for measuring the participant's understanding of the agent. However, it can be questioned whether ATCOs are required to make predictions about the algorithm in order to have a healthy collaboration with it. They do not necessarily have to make changes to the agent itself. Chen et al. [7] state that not necessarily all levels of SAT should always be supported.

There was a non-significant trend that Group A was faster in the top-down approach then Group B. Note that Group started with the top-down scenarios. This corresponds with the general idea that top-down approaches requires prior-knowledge for effective usage. This has also been indicated by other research of ecological interfaces [39]. In ATC operation, ATCOs have plenty of opportunity to practice with the interface, agent and task making prior-knowledge less of an issue.

Participants indicated that the top-down approach presents information more efficiently and is less prone to clutter. This corresponds with EID, which supports top-down processing, because of its efficiency in problem-solving [23]. During ATC operation, this can have important advantages. The bottom-up approach might still be used for training. Participants indicated that it follows the logic of the algorithm better and therefore can have benefits in creating a mental model of the agent.

An important limitation of the study is the focus on the understanding of the agent. This is not the only form of understanding that is required for an ATCO. SA of the domain remains important, especially if automation fails and ATCOs have to step-in. The global SA of participants was not measured in the experiment. Participants were focused on their task of understanding automation and did not zoom out to understand the overall situation. Even though the ecological approach cannot be directly compared to an agent-based approach in this research, the limited ability of participants to fully understand the situation gives an indication that an agent-based approach might result in less domain knowledge. Future research should focus on how agent transparency and domain transparency can be best combined by measuring both forms of understanding. A fair balance needs to be found in which both agent and domain transparency are supported.

Participants faced static scenarios in this research. Obviously, this will not be the case in real-time ATC. The task is highly dynamic: aircraft positions change, aircraft enter or leave the sector, etc. Agents will thus propose new trajectories over time and an ATCO has only limited time to judge trajectories proposed by automation. The time aspect should thus be further researched in order to find the effects on the usage of interface elements related to agent transparency.

Transparency levels were forced in this experiment. However, transparency is not necessary limited by specific levels. Participants indicated that at high levels of transparency, the interface might be overwhelming. Supporting more progressive disclosure might help to solve this problem. Steps in creating an interactive transparency slider have already been made. The slider supports progressive disclosure by showing information only if the information is desired by the operator. Further, research could be conducted to investigate how this balances understanding and workload.

VIII. CONCLUSION

This research proposed an ecological approach to create agent transparency. An interface was designed and divided into four levels of transparency in a top-down and bottom-up approach. These levels were tested in a human-in-the loop experiment with ten participants. It was found that higher levels of transparency significantly increased actual and perceived understanding measured by self-confidence.

The top-down approach performed significantly better in questions related to the strategy of automation, while the bottom-up approach was found more useful for making predictions about the agent. Indications, although not significant, were found that prior knowledge might be an important factor for efficient use of the top-down approach. Participants indicated that the top-down approach was better able to show decision-making information and could prevent cluttering of the interface as information was presented more efficiently.

Future research should investigate how agent and domain transparency could be combined and should test global situation awareness in addition to understanding of automation. Additionally, the effects of a dynamic work domain should be taken into account in future research.

REFERENCES

[1] SESAR Joint Undertaking. *European ATM master plan*. Publications Office of the European Union, Luxembourg, 2020.

[2] L. Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

[3] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 2018.

[4] A. Holzinger, M. Plass, K. Holzinger, G.C. Crişan, C.M. Pintea, and V. Palade. A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop. *Creative Mathematics and Informatics*, 2017.

[5] A. Bhaskara, M. Skinner, and S. Loft. Agent Transparency : A Review of Current Theory and Evidence. *IEEE Transactions on Human-Machine Systems*, 50(3):215–224, 2020.

[6] M.P. Linegang, H.A. Stoner, M.J. Patterson, B.D. Seppelt, J.D. Hoffman, Z.B. Crittendon, and J.D. Lee. Human-automation collaboration in dynamic mission planning: A challenge requiring AN ecological approach. In *Proceedings of the Human Factors and Ergonomics Society*, 2006.

[7] J.Y.C. Chen, K. Procci, M. Boyce, J. Wright, A. Garcia, and M.J. Barnes. Situation Awareness–Based Agent Transparency. *US Army Research Laboratory*, 2014.

[8] M. Ghazizadeh, J.D. Lee, and L. Boyle. Extending the Technology Acceptance Model to assess automation. *Cognition, Technology and Work*, 2012.

[9] B. Guiost, S. Debernard, T. Poulain, and P. Millot. Task allocation in air traffic control involving a common workspace and a cooperative support system. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 9, pages 90–96, 2006.

[10] M. Eiband, H. Schneider, M. Bilandzic, J. Fazekas-Con, M. Haug, and H. Hussmann. Bringing transparency design into practice. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pages 211–223, 2018.

[11] H. Cramer, V. Evers, S. Ramlal, M. Van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 2008.

[12] T. Kulesza, S. Stumpf, M. Burnett, and I. Kwan. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *Conference on Human Factors in Computing Systems - Proceedings*, 2012.

[13] J.B. Lyons, G.G. Sadler, K. Koltai, H. Battiste, N.T. Ho, L.C. Hoffmann, D. Smith, W. Johnson, and R. Shively. Shaping trust through transparent design: Theoretical and experimental guidelines. In *Advances in Intelligent Systems and Computing*, 2017.

[14] C. Westin, C. Borst, and B. Hilburn. Automation Transparency and Personalized Decision Support: Air Traffic Controller Interaction with a Resolution Advisory System. *IFAC-PapersOnLine*, 49(19):201–206, 2016.

[15] C. Westin, B. Hilburn, C. Borst, E.J. van Kampen, and M. Bång. Building transparent and personalized ai support in air traffic control. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–8, 2020.

[16] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi. The ethics of algorithms: Mapping the debate. *Big Data and Society*, 3(2):1–21, 2016.

[17] A. Abdul, J. Vermeulen, D. Wang, B.Y. Lim, and M. Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April, 2018.

[18] C. Borst, J.M. Flach, and J. Ellerbroek. Beyond ecological interface design: Lessons from concerns and misconceptions. *IEEE Transactions on Human-Machine Systems*, 2015.

[19] M.M. Van Paassen, C. Borst, M. Mulder, R. Klomp, P. Van Leeuwen, and M. Mooij. Designing for shared cognition in air traffic management. *Journal of Aerospace Operations*, Vol. 2, 2013.

[20] H.R. Idris, T. El-Wakil, and D.J. Wing. Trajectory planning by preserving flexibility: Metrics and analysis. *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–14, 2008.

[21] H.R. Idris, N. Shen, T. El-Wakil, and D.J. Wing. Analysis of trajectory flexibility preservation impact on traffic complexity. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–14, 2009.

[22] H.R. Idris, N. Shen, and D.J. Wing. Improving separation assurance stability through trajectory flexibility preservation. *10th AIAA Aviation Technology, Integration and Operations Conference 2010*, pages 1–11, 2010.

[23] K.J. Vicente and J. Rasmussen. Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):589–606, 1992.

[24] C. Borst. Shared Mental Models in Human-Machine Systems. *IFAC-PapersOnLine*, 49(19):195–200, 2016.

[25] R. Parasuraman, T.B. Sheridan, and C.D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 30(3):286–297, 2000.

[26] K.J. Vicente. *Cognitive work analysis: Toward safe, productive, and healthy computer-based work.* Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US, 1999.

[27] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[28] A. Adadi and M. Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[29] B.Y. Lim, A.K. Dey, and D. Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2119–2128, 2009.

[30] R.F. Kizilcec. How much information? Effects of transparency on trust in an algorithmic interface. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2390–2395, 2016.

[31] A. Springer and S. Whittaker. Progressive Disclosure: Designing for Effective Transparency. *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019.

[32] J. Rasmussen. Skills, Rules and Knowledge: Signals, Signs and Symbols , and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3):257–266, 1983.

[33] M.R. Endsley. Toward a theory of situation awareness in dynamic systems. Human Factors. *The Journal of the Human Factors and Ergonomics Society*, 1995.

[34] A.R. Selkowitz, S.G. Lakhmani, C.N. Larios, and J.Y.C. Chen. Agent transparency and the autonomous squad member. In *Proceedings of the Human Factors and Ergonomics Society*, 2016.

[35] J. Wright, J. Chen, M. Barnes, and P. Hancock. Agent reasoning transparency: The influence of information level on automation-induced complacency. Technical Report ARL-TR-8044, ARL, US Army Research Lab, Aberdeen Proving Ground, Aberdeen, June 2017.

[36] B. McGuinness. Quantitative analysis of situational awareness (QUASA): Applying signal detection theory to true/false probes and self-ratings. In *2004 Command and Control Research and Technology Symposium*, 2004.

[37] D.S.A. Ten Brink. Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control. Master's thesis, 2017. Unpublished.

[38] N. Naikar, R. Hopcroft, and A. Moylan. Work domain analysis: Theoretical concepts and methodology. *Defence Science and Technology Organisation*, 2005.

[39] R. Klomp, C. Borst, M.M. Van Paassen, and M. Mulder. Expertise Level, Control Strategies, and Robustness in Future Air Traffic Control Decision Aiding. *IEEE Transactions on Human-Machine Systems*, 46(2):255–266, 2016.

II

Book of Appendices

A

Literature Report

Note: this part has already been graded under course AE4020

A.1. Air Traffic Control

This chapter gives an overview of how ATC is organized and what the projections are of the future ATM system. This knowledge will help in creating a relevant simulation. This chapter will provide examples of the LVNL and MUAC ATC service providers. However, the simulation used in the experiment will be a custom made design. Section A.1.1 elaborates upon ATC organisation by discussing how ATC is structured. Furthermore, a brief overview of tools used by ATC has been provided in Section A.1.2. Section A.1.3 describes the future of ATM. It is predicted that higher levels of automation will be introduced into the ATM system. As a result the role of the human will change.

A.1.1. Structure of Air Traffic Control

There are three parts of Air Traffic Control services [55]. Aerodrome Control services are responsible for aircraft in and around the airport. They control the Control Zone (CTR). Approach Control (APP) will handle departures and approaches of aircraft around the CTR in the Terminal Control Area (TMA), which is usually up to FL095 and approximately a 50 km radius around the airport. The rest of controlled airspace is left for Area Control (ACC). ACC handles en-route traffic for aircraft crossing the ACC sector. Furthermore, ACC directs aircraft to APP in case the aircraft is planning to land within that sector. In the Dutch civil airspace a distinction is made between control of Upper Airspace (UTA) and Lower Airspace. Upper Airspace starts at FL245. The Control Areas (CTA) in the Lower Airspace is controlled by the Area Control of LVNL in Amsterdam. The Upper Airspace is controlled by Maastricht Upper Area Control. In this research, ATC is limited to only Area Control of en-route traffic, which is just crossing the sector from one waypoint to another. Therefore, for the remainder of this document, the focus will be on an ACC dealing with en-route traffic.

The main responsibilities of ATC is preventing loss of separation between aircraft under its control [55]. Furthermore, ATC should organize and expedite flow of traffic. In order to do so, ATC is able to give instructions, clearances and information to aircraft. An aircraft should be under control of only one ATC unit at any given time. Aircraft need to be handed over when the aircraft leaves the sector. This resembles transferring the responsibility of one ATC unit to another. Alongside the main priority of maintaining safety, ATC also focusses on punctuality, cost-effectiveness and environmental responsibility [22].

In order to be able to make sure no collision will happen, ATC maintains separation between aircraft. ATC units have to make sure that aircraft pass each other without breaking separation minima. ICAO [38] prescribes that the aircraft should either be vertically or horizontally separated from each other. For vertical and horizontal separation a distance of 1000ft and 5NM are the standards respectively. The restricted area around an aircraft can be visualised as a ice-hockey puck.

The ATCO is able to give clearances and instructions to aircraft. Aircraft must then obey these

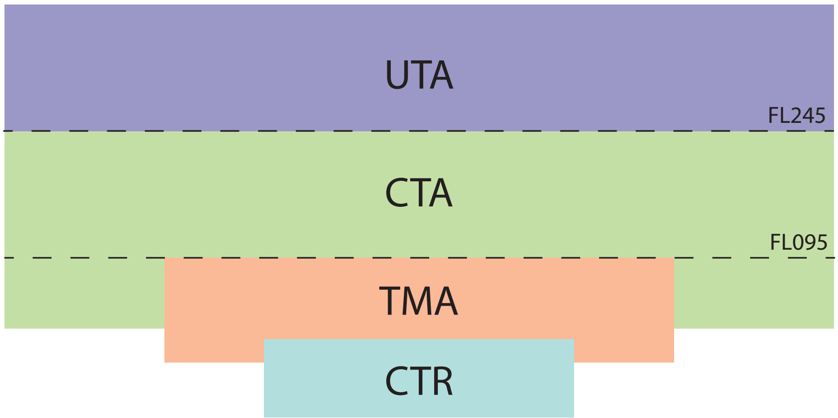


Figure A.1: Overview of controlled airspaces in the Amsterdam Flight Information Region (FIR) [32]

instructions, if they are able to comply with them. If not, they have to inform the ATCO. When clearance or instruction is accepted, it is expected by ATC that the aircraft will do as is agreed upon. In general ATCO has three types of clearance it is able to give to aircraft in order to direct the traffic. These are heading, altitude and speed commands. Heading and speed commands are tools for handling horizontal separation. Altitude instruction can be used for vertical separation.

En-route traffic normally follow their planned route and do not require much ATC interference other than clearances to fly direct to a certain point and changes in speed and altitude. In this research, the focus is on aircraft that have to deviate from a pre-planned route due to a disturbance. The ATCO may add an intermediate point in order to reroute the aircraft instead of a heading or speed command.

Usually ATCO are working alone or in small teams [9]. In the LVNL ACC, the following functions are present [31]:

- **Radar Traffic Controller** : Coordinates the radio communication, executive controller
- **Planning Traffic Controller** : Traffic planning, assists radar ATCO
- **Stack Traffic Controller** : Controls stack in between ACC and APP
- **FIC Traffic Controller** : Provides Flight Information
- **Traffic Control Assistant** : Administrative support

MUAC has a similar distribution of functions. In this research, the focus will be limited to a single Radar Traffic Controller. This controller is the executive controller and has a direct link to the re-routing of aircraft. Automation will thus primarily change his role. Furthermore, it is easier to focus on one controller instead of a group of controllers during the experiment. A single controller requires less participants in total and will result in a more controlled environment without the interference of the effects of teamwork.

A.1.2. ATC equipment and tools

In order to manage traffic, ATCOs make use of equipment and tools. In the beginning, ATC made use of radio and so called “shrimp boats” to keep track of positions of ATC. Nowadays, ATC uses more advanced tools. A brief overview of available equipment and tools for ATC will be given in this section. Note that there is variation in the adoption of advancements in ATC systems per ATC provider and that not all ATC units have state-of-the-art systems. Many facilities develop their own tools for their interfaces. An example is the interface built by MUAC in Figure A.2. So there are many more tools or display elements that could be added to this brief overview. There is no strict standard of the Human Machine Interaction (HMI) system. However, standardizing of HMI systems is one of the improvements suggested in the ATM Master Plan [73].

Figure A.2: MUAC radar Screen [22]

A.1.2.1 Radar screens

The primary source for information for an area control ATCO is the radar display. In Figure A.3, an example of a radar display can be seen. Multiple forms of information are represented in this screen such as: lists, air routes, aircraft positions and weather. A general overview of this will be given in the following paragraphs.

Tracks and labels The radar presents positions of aircraft. These positions are updated at a frequency of 4 - 15 seconds depending on the range of the radar. For each aircraft, the aircraft ID, actual and cleared altitudes, ground speed, heading, exit waypoint and aircraft type are usually shown on a label accompanying the aircraft position indicator. An example of a label can be seen in Figure A.4. The past positions are referred to as history dots and can also be seen in Figure A.4.

Contextual information In addition to aircraft position and information, additional information about the context is displayed on the radar screen as well. One example is the map of the controlled area including restricted airspaces and obstacles. Other examples are route displays that show the commonly flown airways in the airspace. Also external factors such as the weather could be integrated.

Tools: STCA, MTCD and TCT Short Term Conflict Alert (STCA), Mid Term Conflict Detection (MTCD) and Tactical Controller Tool (TCT) are tools to prevent collisions. The differences between them are shown in the Table A.1. Note that the time at which they predict loss of separation differs. STCA typically has a look ahead time of two minutes, whereas MTCD has a time horizon of up to 30 min. TCT is somewhat in between STCA and MTCD and typically has a look ahead time of 5 to 8 minutes. It thus may not come as a surprise that all system are used for different purposes. STCA is used as a safety net to warn the controller for an imminent breach of separation. The MTCD is used for planning traffic in advance. It will warn the controller of potential conflicts due to controller clearances and the flight plan. Finally the TCT is used for conflict resolution and clearance verification. It checks whether current clearances are correct or are in need of correction.

Table A.1: Differences between the three conflict detection tools STCA, TCT and MTCD [66].

	STCA	TCT	MTCD
Look ahead time	1-2 min	5-8 min	20 - 30 min
Input	Surveillance data	Surveillance data + flight plan data	Flight plan data + controller input
Used as	Safety net	Conflict resolution	Planning tool

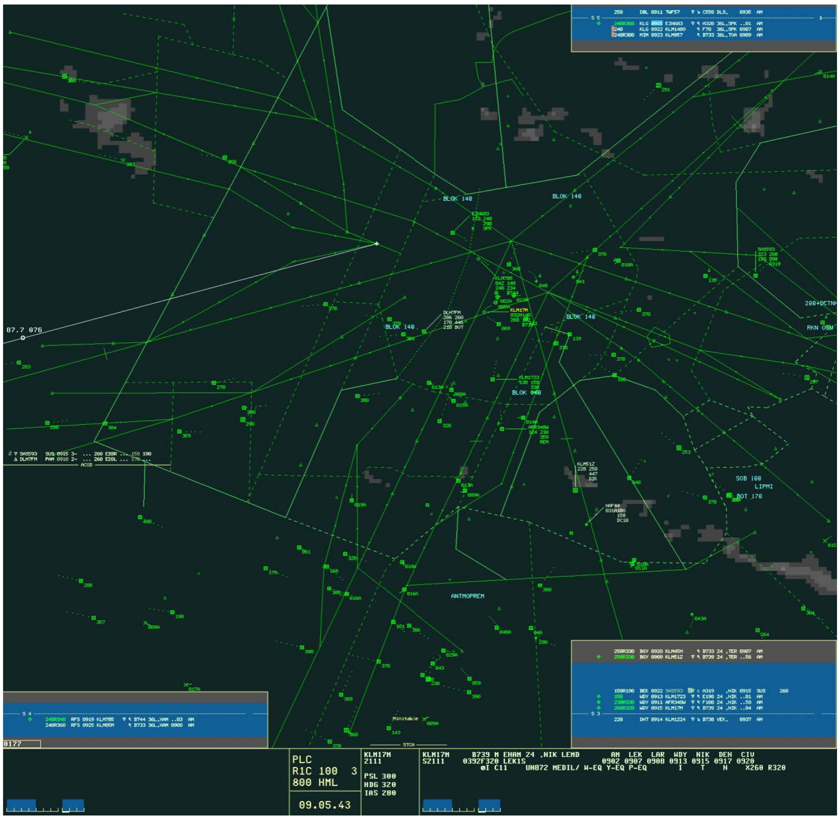


Figure A.3: Example of a ACC radar screen [31].

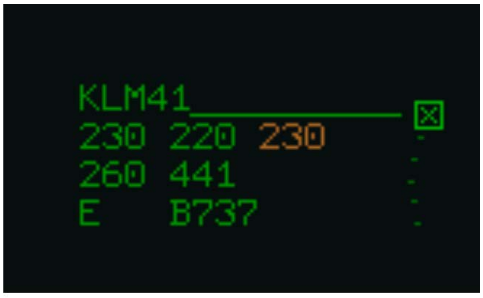


Figure A.4: Aircraft track symbol (box with cross), label (left) and history dots (small dots underneath track symbol) [31]

Flight strips Physical flight strips have been, and in many ATC providers still are, successful tools for tracking flight. Flight strips allow controllers to instantly grasp relevant information of the aircraft. Furthermore, handing over flight strips literally resembles handing over responsibility. Nowadays, flight strips are becoming electronic instead of physical. Some ATC providers are even moving away form flight strips by placing all relevant information in other places on the screen [65]. Aircraft labels on radar screen are an example of moving critical information from flight strips to the screen.

Lists Another common type of information on radar screens is lists. The most known list on the screen is the list of flight progression strips as is discussed above. Other lists for ACC are entry lists. These lists contain all flight that will enter the airspace at that specific entry point. In addition lists of aircraft in close conflicts with each other are displayed in the STCA list. Similarly the Area Conflict Detection (ACOD) lists conflicts of aircraft with restricted areas.

A.1.2.2 Communication: radio and CPDLC

Communication between pilot and controller is another important part of ATC. This has been done for many years using radio communication and as of today radio is the primary form of communication between aircraft and ground. For radio communication it is important that radio discipline is followed. Nowadays, communicating via CPDLC is becoming globally implemented [64]. Controller Pilot Data Link Communication (CPDLC) is used for non-urgent strategic messages. These message could be clearances, information or requests and are phrased using voice phraseology of ATC procedures. Introducing CPDLC has benefits of decreasing communication on ATC frequencies as well as increasing sector capacities and reduced probability of miscommunication. However, downsides exist such as less benefit of the so called “party line effect”, which gives pilots awareness of intentions of other aircraft.

A.1.3. Future of ATM

The air traffic management system of today is going to change. The ATM Master Plan describes the necessity of developing the new air traffic management system [73]. According to these documents, it is anticipated that by 2035 the European airspace will have to handle over 17 million flights annually [73]. For comparison, the number of flights in 2018 was close to 11 million. Furthermore, due to ATC capacity and staffing constraints, the Single European Sky (SES) delay target of 0.5 minutes/flight has not been met since 2015. A related problem is that increasing capacity of ATC is getting more and more complex and costly.

The SESAR vision is to deliver a scalable air traffic management system, which is able to handle growth in air traffic. Therefore, many of the tasks done manual in aviation today will be automated in the future in order to increase the scalability and safety. Aviation should digitally transform characterised by a significant increase in levels of automation. It is mentioned that the goal is not to automate for the sake of automation, but rather to optimize the socio-technical ATM system and increase human performance and involvement within that system.

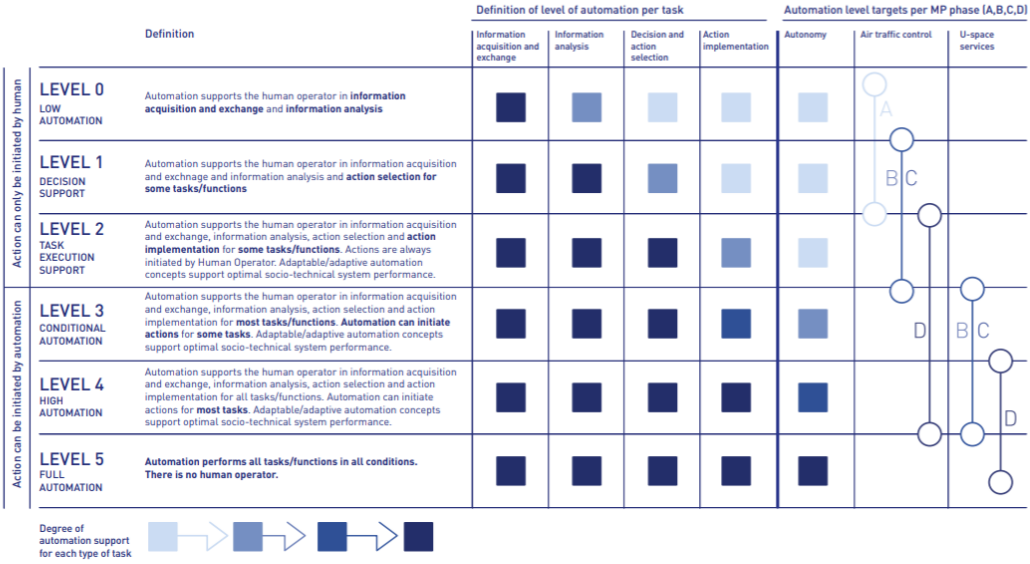
Automating parts of air traffic control could have many benefits. The airspace could be used more efficiently, which increases the capacity of the airspace. Another benefit is better environmental and cost performance, which could be reached by flying more direct routes for example. Automation could also benefit human performance through, for example, workload reduction of ATCOs who are managing busy airspaces.

A.1.3.1 ATM automation

How will the ATM system be automated? The ATM Master Plan [73] gives an overview of which parts of the ATM system will be reorganised or automated. In a top level view, the ATC and ATFM will slowly merge during the planned increase in automation. The expectation is that automation will take on more and more tactical ATC tasks. This is necessary in order to cope with the planning versus flexibility paradox [16]. As more of the system is planned, it becomes increasingly difficult to deal with unexpected events due to less freedom. Introducing more automation or automation support in tactical air traffic control will allow for more planning and thus a more strategic mindset. This paves the way for more advanced ATFM methods that rely on increased automated tactical control of aircraft.

Hybrid human-machine teaming is a central concept in automating tactical control. Task allocation should be guided by advanced adaptable and adaptive automation principles. For tactical control, automation will mean that tasks will be delegated from the ATCO to machines. These machines will then propose solution to the human in order to help make it more efficient and safe. Especially in the presence of complex trajectories or high density traffic situations, proposing solutions, and maybe even executing these solutions, could improve system performance and resilience.

Automating systems will be gradual. The ATM Master Plan has four phases. In Figure A.5, phases A-D are shown in combination with the increase of level of automation for ATC. In phases A-C, the focus will be on increasing system support. Humans will still initiate action in these phases, but they will be supported by enhanced automation. In phase D, this changes and the human will be removed from the control loop for selected ATC tasks. In this phase, it is important that collaboration between the human and the automation is facilitated as well. This research fits best in levels 3 and 4. In these levels automation is able to initiate and suggest action. However, there is still a (supervising) human controller. In these situations, agent transparency becomes critical for successful human-machine collaboration [8].



			Definition of level of automation per task				Automation level targets per MP phase (A,B,C,D)		
		Definition	Information acquisition and exchange	Information analysis	Decision and action selection	Action implementation	Autonomy	Air traffic control	U-space services
Action can only be initiated by human	LEVEL 0 LOW AUTOMATION	Automation supports the human operator in **information acquisition and exchange** and **information analysis**	█	▒	░	░	░	A	
	LEVEL 1 DECISION SUPPORT	Automation supports the human operator in information acquisition and exchnage and information analysis and **action selection for some tasks/functions**	█	█	▒	░	░	B C	
	LEVEL 2 TASK EXECUTION SUPPORT	Automation supports the human operator in information acquisition and exchange, information analysis, action selection and **action implementation** for **some tasks/functions**. Actions are always initiated by Human Operator. Adaptable/adaptive automation concepts support optimal socio-technical system performance.	█	█	█	▒	░		
Action can be initiated by automation	LEVEL 3 CONDITIONAL AUTOMATION	Automation supports the human operator in information acquisition and exchange, information analysis, action selection and action implementation for **most tasks/functions**. **Automation can initiate actions** for **some tasks**. Adaptable/adaptive automation concepts support optimal socio-technical system performance.	█	█	█	▓	▒	D	B C
	LEVEL 4 HIGH AUTOMATION	Automation supports the human operator in information acquisition and exchange, information analysis, action selection and action implementation for all tasks/functions. Automation can initiate actions for **most tasks**. Adaptable/adaptive automation concepts support optimal socio-technical system performance.	█	█	█	█	▓		D
	LEVEL 5 FULL AUTOMATION	**Automation performs all tasks/functions in all conditions. There is no human operator.**	█	█	█	█	█		

Degree of automation support for each type of task ░ → ▒ → ▓ → █

Figure A.5: Levels of automation for air traffic control for phases A-D. Retrieved from [73]

A.1.3.2 The role of the human

With these plans for increasing the level of automation, one can only conclude that the role of humans in the system will inevitably change. Tasks and responsibilities traditionally assigned to human operators will partially be taken over by machine actors. The strength of humans to handle the unexpected should be used optimally. However, the traditional believe that humans are able to do this unaided and unsupported is no longer seen as valid. Humans will be assisted by machines to manage these situation quickly and safely [73].

Delegating tasks to machines, will relieve the human of some tasks. However it should not be assumed that the role of the human will become easier. The operator will have to perform differently in comparison to the current system and in many cases this will be a more demanding job [16]. Therefore, development of new HMI interfaces is necessary to decrease both mental and physical workload. Examples of these new modes include: in-air gestures, attention control, user profile management systems, tracking labels, virtual and augmented reality and more [73].

New tools are required for the human in order to be able to succeed in this new role. The SESAR Target Concept [16] has developed a set of high-level automation principles in order to align automation development with their vision. The relevant principles for this project are stated below:

- Automate only to improve overall system and human performance, not just because the technology is available

- Examine the overall impact of automation before implementation to avoid additional complexity, loss of appropriate situation awareness or increase of errors;
- Place the human in command. The human will be the automation manager and not the automation monitor. Automation will assist humans to carry out their tasks safely, efficiently and effectively. Furthermore, the delegation of authority to machine should be clearly defined in all operational situations;
- Involve users in all phases of system design to ensure, inter alias, benefits for overall system performance and to foster trust and confidence in the automation functions;
- Consider the respective typical strengths and weaknesses of humans and of technology when deciding what to automate.

The change in role of the human also indicates that the knowledge and skills have to change. Generally, tasks will be more managerial and complex in comparison to the tasks of today. This requires more training. Also more in depth-technical knowledge is required due to the added level of complexity to the system. Agent transparency could be used to support the human in acquiring this knowledge of the system.

A.2. Human-Automation Interaction

Further automation in ATC is inevitable as described in Chapter A.1. Increasing the level of automation in the system will undoubtedly have an effect on the role of the human. This chapter deals with the interaction between human and automation. In section A.2.1, the ironies of automation are discussed. These show the pitfalls that automation could bring along. In section A.2.2, Billings' Human Centered Automation (HCA) is described. HCA and the ironies of automation substantiate the relevance of this research. It is briefly described how the interface corresponds to the principles of HCA and how it could counteract some of the ironies of automation. In section A.2.3 a taxonomy for the levels of automation is shown. This taxonomy is used as a communication framework. Finally, Section A.2.4 describes how ecological interfaces can be created. It discusses the design philosophy of Ecological Interface Design (EID) and reviews the cognitive work analysis, which will form the basis of EID.

A.2.1. Ironies of automation

In 1982 Lisanne Bainbrigde described the ironies of automation [5]. These ironies support the paradox that increasing the automation will make the role of the human operator more important instead of less. Humans are pushed to do two different types of tasks according to Bainbridge. Due to automating processes the human will have to monitor the automation instead of acting him/herself. Furthermore, a human operator needs to be able to intervene when automation fails. Therefore, the ironies are mainly focused on monitoring and control activities. The ironies are:

1. Designers think that humans are inefficient and unreliable. However, designer errors create operational problems in automated systems. Furthermore, the designer leaves tasks that cannot be automated for the human operator.
2. Physical skills of human operators deteriorate due to lack of practice. However, an operator needs to be more skilled instead of less if automation fails and the operator needs to take control.
3. Human operator is pushed to a monitoring role. A role in which humans do not perform well.
4. Information about the system is harder to obtain in monitoring than in active participation and thus the operator needs to take decisions on a minimal amount of information.
5. Automation makes the system more complex instead of less. The extra layer of automation can hide automation degradation.

One might question whether these ironies have not been solved already. Meanwhile the article of Bainbridge was published almost 40 years ago. According to Baxter et al. [7], these ironies are still very much alive. It was stated that automation in aviation made it more difficult to predict the outcome of actions and pilots have increased understanding issues about the behaviour of the aircraft. These problems relate to Ironies 4 and 5. In their analysis, human operators are the last defence if automation fails. The following three solutions to prepare humans were proposed:

1. Gain skills
2. Update and practice skills
3. Provide appropriate information by the technology in a timely manner.

Creating interfaces that are able to provide relevant information of the inner-workings of the algorithm could help the operator predict (and supervise) action outcomes as well as increase the understanding of system behaviour. Therefore, the interface could help counteract Ironies 4. Furthermore, potential flaws of automation, Irony 1 and 5, could be detected due to the information presented on the interface. The interface will not help solve Ironies 2 and 3. The interface will not necessarily keep the human out of the monitoring role, or train the operator in order to counteract skill degradation.

A.2.2. Human Centered Automation

An interesting design philosophy regarding automation is that of Charles Billing. In his book [9], he proposes a set of principles that constitutes Human Centered Automation. This set of principles is based on the idea that humans have the end responsibility and therefore have to be in command. The human should be the primary focus in designing automation. Tools are auxiliary and are there to help the human. Billings' principles of Human Centered Automation are:

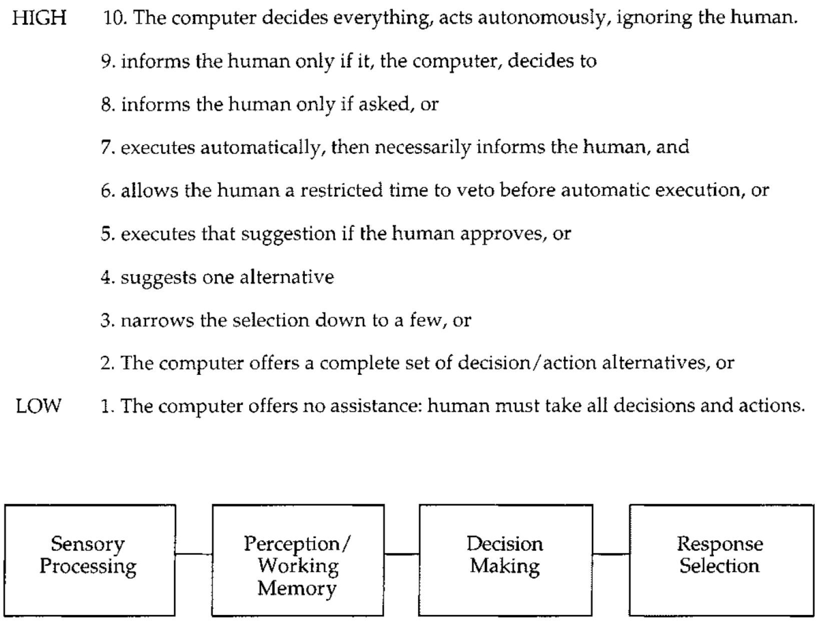
1. Operator must be actively involved.
2. Operator must be adequately informed.
3. Operator must be able to monitor the automation assisting him.
4. The automated system must therefore be predictable.
5. The automated system must also monitor the human operator.
6. Every intelligent system element must know the intent of the other intelligent system elements.

Note that this is a design philosophy and not a design framework or method. It does not explain how a human-machine interface should be developed. However, it shows what automation should offer us and explains why it should offer it. Furthermore, it is good to question the design to check whether it fits these principles. As Billings explains in his book, these rules are not set in stone. They are flexible and the engineer should re-evaluate them for each design and choose whether to adhere to them or not. Therefore, testing the design with these principle enables substantiation of the design.

So how does this design philosophy relate to this research? The majority of these principles are applicable. Only principle 5 is of minimal interest. The main focus will be on the last principle. The aim of the to-be-developed interface is to make the inner working of the algorithm transparent to the operator. The idea is that the increased transparency increases the understanding of the operator and it helps him discover the intentions of the computer agent as well as predict its outcomes. This will enable the operator to monitor the automation more effectively and will support the operator in explaining the algorithm's reasoning.

A.2.3. Levels of automation

An important concept in the field of human-automation research is the levels of automation (LOA). The LOA create a division of roles and responsibility between humans and automation [12]. This is done by allocating authority and autonomy to human and machine. For this research, the LOA is used as a communication framework. Therefore, only one of the many taxonomies will be presented in this section.



HIGH 10. The computer decides everything, acts autonomously, ignoring the human.
9. informs the human only if it, the computer, decides to
8. informs the human only if asked, or
7. executes automatically, then necessarily informs the human, and
6. allows the human a restricted time to veto before automatic execution, or
5. executes that suggestion if the human approves, or
4. suggests one alternative
3. narrows the selection down to a few, or
2. The computer offers a complete set of decision/action alternatives, or
LOW 1. The computer offers no assistance: human must take all decisions and actions.

Figure A.6: The LOA taxonomy proposed by Parasuraman, Sherida and Wickens [56].

Figure A.6, shows the 10 levels of automation as proposed by Sheridan and Verplank [63]. The lower 5 levels resemble the levels of automation in which the human is still in control. The upper 4 levels describe a system in which the machine has control. Parasuraman, Sheridan and Wickens [56] have expanded this taxonomy by applying the LOA to 4 broad classes of functions: information acquisition, information analysis, decision selection and action implementation.

During this research, automation will have LOA level 5 on decision support. This will be equivalent to “management by consent”. Once an aircraft enters the airspace, automation will be activated and proposed a solution to the human. The solution will be executed if the human approves.

A.2.4. Ecological Interface Design

Ecological Interface Design (EID) was introduced by Rasmussen and Vicente [78]. EID differs from other approaches [23]. In User-Centered Design the focus is on the limitations and capabilities of humans and their application to design. Technology-Centered approach is focused on the limitations and capabilities of the technologies. In EID, the focus shifts from the interaction between humans and machine, to the interaction between humans and work. Human and technological limitations should still be considered, but they should be seen in the larger context of the work ecology. Ecological Interface Design makes use of the so called Triadic Approach; see Figure A.7.

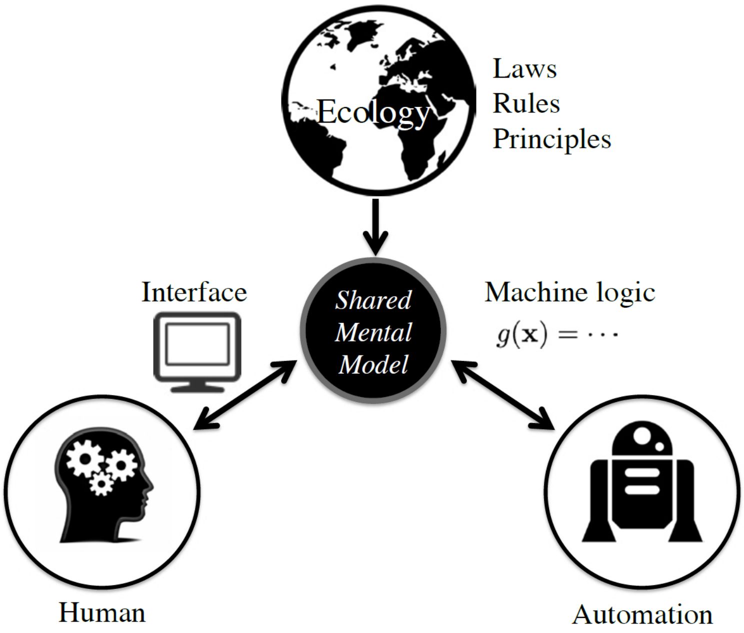


Figure A.7: The triadic approach [10].

In EID the meaningful relationships and constraints in the work domain should be visualised in the interface. Therefore, EID helps in creating domain transparency. In this research, the EID framework is used to also visualize the meaningful relation between the agent and its work domain. It is assumed that the agent is part of the work domain itself. The relationships that it has with the work domain will then become more salient and can be used to identify important information for creating agent transparency.

In order to show the meaningful relationships, it mainly deals with two questions. The first one being: "What is the right information?" and the second question is: "How to communicate the right information?". The Cognitive Work Analysis (CWA) and the Skill- Rull and Knowledge (SRK) Taxonomy will form the basis for answering both questions. The next two section will describe both theories. The last section will provide two examples of ecological interfaces that try to visualize the work domain.

A.2.4.1 SRK Taxonomy

The SRK Taxonomy of Rasmusssen [61] is used to differentiate between behaviours. In EID, the goal is to support all behaviour types. This taxonomy describes three types of behaviours.

SBB The Skill Based Behaviour (SBB) is primarily unconscious. These are actions directly coupled to the environment. They are automatic responses of the neuromuscular system to signals of the environment. Reflexes are an example of SBB.

RBB Rule Based Behaviour (RBB) is considered with executing stored rules (if-then). These could be obtained by following protocol or are learned by experience. In contrast with SBB, RBB is cognitive. Therefore, people are able to explain this type of behaviour. The system goals are not considered during RBB. The operator is reacting based on recognized cues (signs) and executes the stored rule that corresponds with these signs.

KBB Knowledge Based Behaviour (KBB) is characterised by considering the system goal explicitly. In comparison to RBB and SBB, KBB is slow, serial and effortful. During KBB the operator needs to interpret the meaning of symbols and make decisions that correspond to the system goals.

Training affects these types of behaviours. KBB could become RBB after the operator has encountered the same situation multiple times. Even RBB could turn into SBB after extensive training on a particular situation. Novice operators usually start at KBB and will then create rules and skills for RBB and SBB during training [39].

A.2.4.2 Theory of Cognitive Work Analysis

The cognitive work analysis (CWA) will form the basis of the EID. The CWA framework was introduced by Vicente [77]. It consists of 5 parts that together describe the work that needs to be done with respect to its domain and actors. A description of all steps is provided in this section. In Chapter A.5, the CWA for this research is performed.

Work Domain Analysis The work domain analysis is used to define the task environment [39]. This phase starts with stating the system boundaries in order to isolate the work domain. Then, it identifies the fundamental constraints, both physical and purpose related, that drive the work domain. These constraints are irrespective of the specific task or actor. This helps in creating a shared understanding or common ground between the actors, which corresponds with the Human Centered Automation philosophy as described by Billings [9].

The Abstraction Hierarchy (AH) is the main tool that is used in work domain analysis. The AH makes a connection between the physical basis and the purpose using the relationships between different levels of abstraction. The AH has 5 different levels of abstraction. At each level of abstraction, the level above and below are connected with a means-end relationship as can be seen in Figure A.8. They answer the Why-What-How questions. The current level describes “What”, the level above gives a reason and explains “Why”, the level below gives a clue on “How”.

Therefore, the AH is used to answer the first question in EID: “What is the right information?”. The AH makes the relationships within the work domain salient and is therefore a basis for both the information content and structure [78].

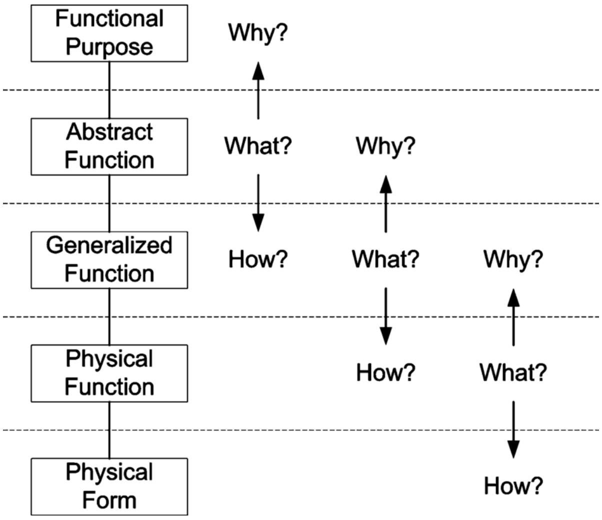


Figure A.8: Means-end relationship in Abstraction Hierarchy. Retrieved from [46]

If the AH is used for identifying information for creating agent transparency, it is interesting to include the agent into the system boundary. The advantage is that the relationships between the elements of the actor and other elements in the work domain become salient. Even though including the actor into the system boundary is controversial, it is not unprecedented. An example of other technology that is integrated into the AH is aircraft [11]. It can be argued that the rerouting agent is a technology acting in the work domain and that this is similar to aircraft that also act in it.

Control Task Analysis In control task analysis, the goal is to create a better understanding of the task at hand [39]. It identifies what needs to be done and all steps that are undertaken during the control task. These steps are independent of the actor that will perform the task. Vicente [77] proposes to use the decision ladder [60] to identify what needs to be done. The steps in the decision ladder are the activities a novice would make to carry out the task. The decision ladder shows information processing activities and states of knowledge. Novices are expected to climb up and down the complete decision ladder. More experienced users can shortcut parts of the decision ladder. Also enhanced interfaces could support users to make shortcuts. In Figure A.21, the decision ladder of this research can be found.

Strategy Analysis The control task analysis states the activities that need to be performed in order to complete the task. The strategy analysis provides insight into how these activities need to be performed [39]. A large variety of strategies should be identified during strategy analysis, because agents might perform the task in a different way. Particularly, the strategy of the automated agent will be of interest as this can give clues for structuring the information that will be presented to the human operator.

Beforehand, it is hard to predict when a strategy would be useful. Additionally, ATCOs tend to switch between strategies in order to keep cognitive workload at an acceptable level [68]. The interface should therefore support seamless transition between the strategies. Designing for the “one right way” should be avoided. The goal is to facilitate a large variety of strategies [77]. The strategy analysis can be visualised using a simplified flow maps in which all strategies to perform the activity are noted and all steps per activity are stated [3]. In Figure A.22, the strategy analysis flow diagram of this research can be found.

Social Organisation and Cooperation Analysis During social organisation and cooperation analyis the tasks are divided between agents. The goal is to find out how agents can best work together in order to improve system performance [39]. It makes clear what the operator needs to do during operation and how automation needs to be designed. This analysis usually does not have a separate tool. The diagrams of the previous phases are colored according to the actor to which they belong.

Worker Competence Analysis During the worker competence analysis, the goal is to find out how the operator can be best supported in his or her task. For answering this question, the SRK taxonomy [61] is used. The lower levels of cognitive control, which are associated with SBB and RBB, can be executed quickly, more effectively and with less effort. EID aims to support these levels of control, while still being able to present information for KBB in case problem solving is required [78]. Thereby, the worker competence analysis helps in determining what elements on the interface are required in order to support all types of behaviour during all sub-tasks.

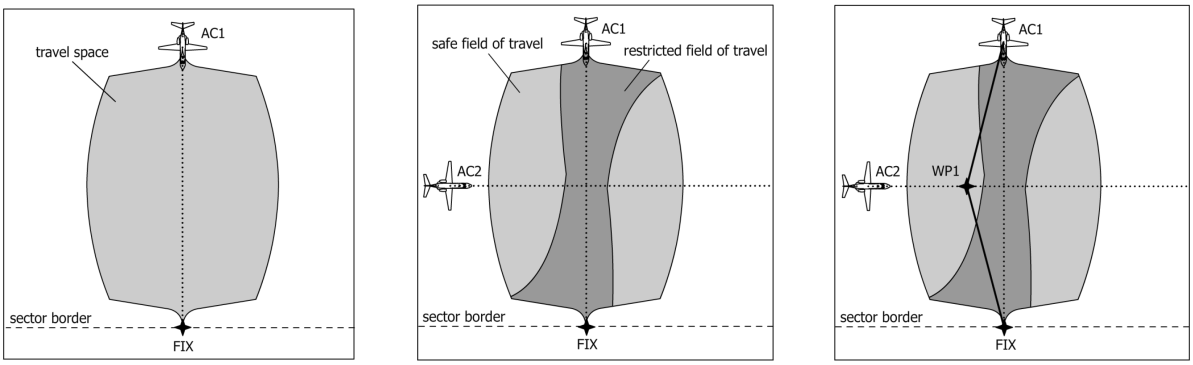
A.2.4.3 ATC Displays Developed

In this section, some examples of ecological displays that try to visualize the work domain will be discussed. Two important displays are the Travel Space Representation (TSR) and the Solution Space Diagram (SSD). These displays support the operator in re-routing of aircraft. They do not necessarily give explanations about the inner workings of automation. However, they support shared cognition and are able to show the rationale behind automation and thereby provide domain transparency.

A.2.4.4 TSR

The ultimate goal of the Travel Space Representation is to [75, p. 39] “design a shared representation that underlies both the design of the human-machine interface and the rationale that guides the automation.” A shared representation would lead to shared cognition between operator and automation. The TSR was created for 4D trajectories. Trajectories that are not only space-bound (3D), but also time bound making the problem four dimensional. The representation shows what areas are available for intermediate waypoint placing while adhering to the constraint given by the 4D trajectory, preventing loss of separation with other aircraft and avoiding restricted areas.

Figure A.9 shows TSR representation. The form of the TSR is based upon a set of ellipsoids and is cut-off in the end and beginning due to turning constraints. Each ellipsoid is determined by a speed. The outer ellipsoid corresponds to the maximum speed of the aircraft. Placing waypoints further away from the direct trajectory will require flying faster in order to arrive at the end point at the Required Time of Arrival (RTA). In Figure A.9b, the areas of the restricted and safe field are indicated with respectively dark and light grey. Conflict resolution was performed by placing a waypoint in the safe field of travel as can be seen in Figure A.9c



(a) TSR of AC1

(b) TSR of AC1 effected by AC2

(c) Conflict resolution for AC1

Figure A.9: TSR overview. Retrieved from [59].

The representation in Figure A.9, is a basic composition of the TSR. More information could be added such as the ellipsoids that resemble flying speed, which will give a clue regarding efficiency. Furthermore, a design was proposed that visualizes the robustness of the safe field of travel [59].

In experiment evaluation of the TSR it was found that the tool never suffered from breakdown and only a very limited amount of safety critical events occurred. The workload for the subjects did increase, but remained on a manageable level. The tool was found to be supportive for the task at hand [43].

The TSR shows the “sea of possibilities”, which allow for safe system performance and will not prescribe predetermined strategies and solutions. This enables the operator to implement their own strategy for solving conflicts. The interface will be most effective for expert operators as they are better in balancing safety and efficiency. Novice operators tend to seek the boundary of safe operation more often [44].

The TSR is able to show the rationale that automation is using as it gives insight about the constraints imposed by the work domain. The abstraction hierarchy of the TSR shows these constraints that the TSR is visualizing. In Figure A.10, this abstraction hierarchy is shown. As will be discussed in chapter A.3, the elements and links in this hierarchy are also used by the Flexibility Metric algorithm. Therefore, the shared representation of the TSR forms a good starting point for explaining decisions of automation. However, it is still limited in the sense that it is not able to show how the algorithm is interpreting the work domain and how it uses that interpretation to come up with a solution. A good starting point for designing an interface for internal transparency would be adding this interpretation of the work domain and usage of the interpretation for decision making.

A.2.4.5 SSD

The Solution Space Diagram (SSD) was initially developed as a decision support tool for pilots to show the travel possibilities [74]. An example of a SSD can be seen in Figure A.11 . Similarly to the TSR, the SSD shows the go and no go areas of the selected aircraft. However, the SSD shows the possible vectors that will result in a conflict-free resolution in contrast to the TSR, which shows the area for placing a intermediate waypoint. A vector in the SSD is a combination of speed and heading.

The display was experimentally evaluated by Mercado Velasco et al [50]. In the experiment two different levels of traffic density were tested in order to test the effect of the SSD on controller workload. The subjects had to merge traffic using the SSD. It was shown that the SSD had significant effects on the workload. This was especially the case for high traffic density scenarios.

Westin et al. [80] used a SSD with only a heading band for investigating the effect of strategic conformance on acceptance of automation. In their research, they replayed the subjects’ own conflict resolution as automated advisories, which the controller could either accept or reject. The idea was that the SSD would provide information for the controller to validate the automation advisory. In 25% of the cases, controllers would disagree with their own conformal advisories.

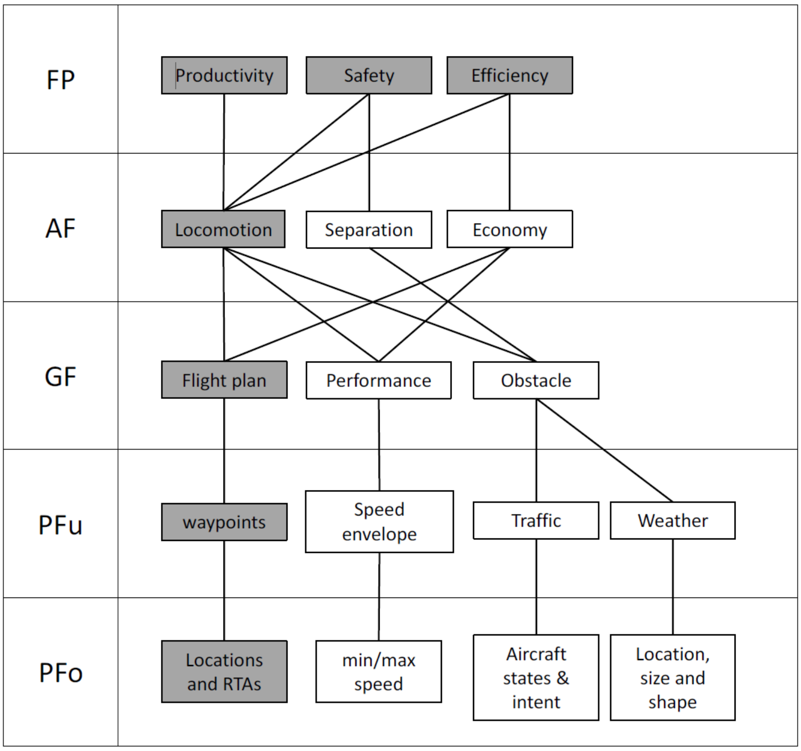


Figure A.10: Abstraction Hierarchy of the TSR [71]. Grey shaded blocks are directly visible, while the other blocks become visible after selecting an aircraft.

In [82], this is explained as a lack of transparency of automation and it is suggested that increasing the transparency would have a positive effect on decision effectiveness, efficiency and predictability of automation. Rather than explaining why a certain decision is suggested, they wanted integrate more information in the SSD interface resulting in a SSD similar as in Figure A.11. Unfortunately, no significant effects of increased transparency were found. However, the new display was found to facilitate understanding of the advised resolution better.

Note that also here explaining the rationale of the automation is limited to increasing the transparency of the work domain. Also for the SSD, it would be interesting to show the interpretation of the work domain by the algorithm and to show its decision making process.

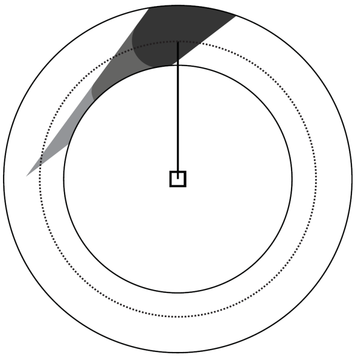


Figure A.11: Example of a SSD with color-coded no-go zones providing information on separation loss proximity [25].

A.3. The Flexibility Metric Algorithm

The previous chapter discussed ecological interfaces that support operators in re-routing aircraft. If a higher level of automation is desired for this task, an algorithm needs to be selected that supports the human in re-routing. In this research, the Flexibility Metric algorithm was chosen to perform this task, because it has a similar approach as the TSR. This chapter will explain the workings of the algorithm. It deals with question such as: How does it work? What are its limitations? And what are important aspects that should be explained? The answers to these questions give good indications about “what to explain” and “how to explain”; This is a starting point for identifying information that is relevant for making the algorithm more transparent.

A.3.1. Algorithm workings

The Flexibility Metric algorithm was chosen, because of its similar approach to the problem as the TSR. Therefore, the newly developed interface that will increase agent transparency, can be better combined with the TSR. Due to its similarity, it is not necessary to explain the difference in approach. In other words, adding agent transparency to the domain transparency will become easier to implement.

This similarity in approach can also be seen in the CWA. There is a large overlap of the constraints of the TSR and the Flexibility Metric algorithm. Both algorithms control the aircraft’s routing by placing an intermediate waypoint. Furthermore, they are using the performance of the aircraft to investigate the area that it is able to be reached: the travel space. Subsequently, they are excluding re-routing options that would result in conflicts. This will lead to a collection of feasible re-routings options. From these options, an optimal trajectory can be chosen that corresponds with the high level goals of the work domain. If this is compared to other path planning algorithms such as Dijkstra [19] or A-star [30], it becomes clear that they have less overlap. Dijkstra or A-Star will only find the optimal solution as it stops when the goal has been reached. This is a clear difference with the TSR approach in which all feasible trajectories are discovered.

The Flexibility Metric algorithm is a Node-Based-Optimal Algorithm. It is developed by Idris et al. [35–37]. Node-Based-Optimal algorithms create a field with nodes and will try to find the most optimal route along those nodes from start to finish. One of the advantages of Node-Based-Optimal algorithms is that they are understandable by humans. Furthermore, Node-Based-Optimal algorithms tend to perform relatively well in terms of computational expense [71]. The Flexibility Metric algorithm has already been implemented in JAVA for an ATC simulator by Ten Brink. He used this algorithm for his research and a detailed explanation about the algorithm was provided by him [71]. The text below is based upon his work and expanded for better comprehensibility where necessary.

The tasks of the algorithm is to find an optimal conflict-free trajectory from the aircraft current position to the exit point of the airspace sector without arriving late at that point. The optimality of the solution depends on the strategy used by the algorithm.

The general idea behind the Flexibility Metric is that controlling the trajectory flexibility of one aircraft will allow the management of the overall traffic complexity, in order to keep it on a acceptable level. The algorithm is able to optimize for multiple strategies such as shortest path, robustness, adaptability and a combination of the three. An overview of the algorithm can be seen in Figure A.12. The information flow is indicated by the arrows. Circles represent information states and the blocks indicate a process. The figure will be used to explain the whole algorithm step-by-step. There are four important steps. In the next sections, these steps are further described. The four steps are:

1. Define metric grid,
2. Find all reachable cells: Forward propagation,
3. Find all possible trajectories: Backward propagation, and
4. Evaluate trajectories and find the best one.

The first three steps are iterated through for a range of time intervals. If no solution was found, the number of segments will be increased. The delay will be increased by relaxing the RTA if increasing the number of segments did not yield any solution.

A.3.1.1 Defining the metric grid

The first step in the algorithm is defining the metric grid. The metric grid consists of a grid strategy, obstacle list, state change strategy and interval strategy. Together, they form a metric grid that forms the bases for further operations.

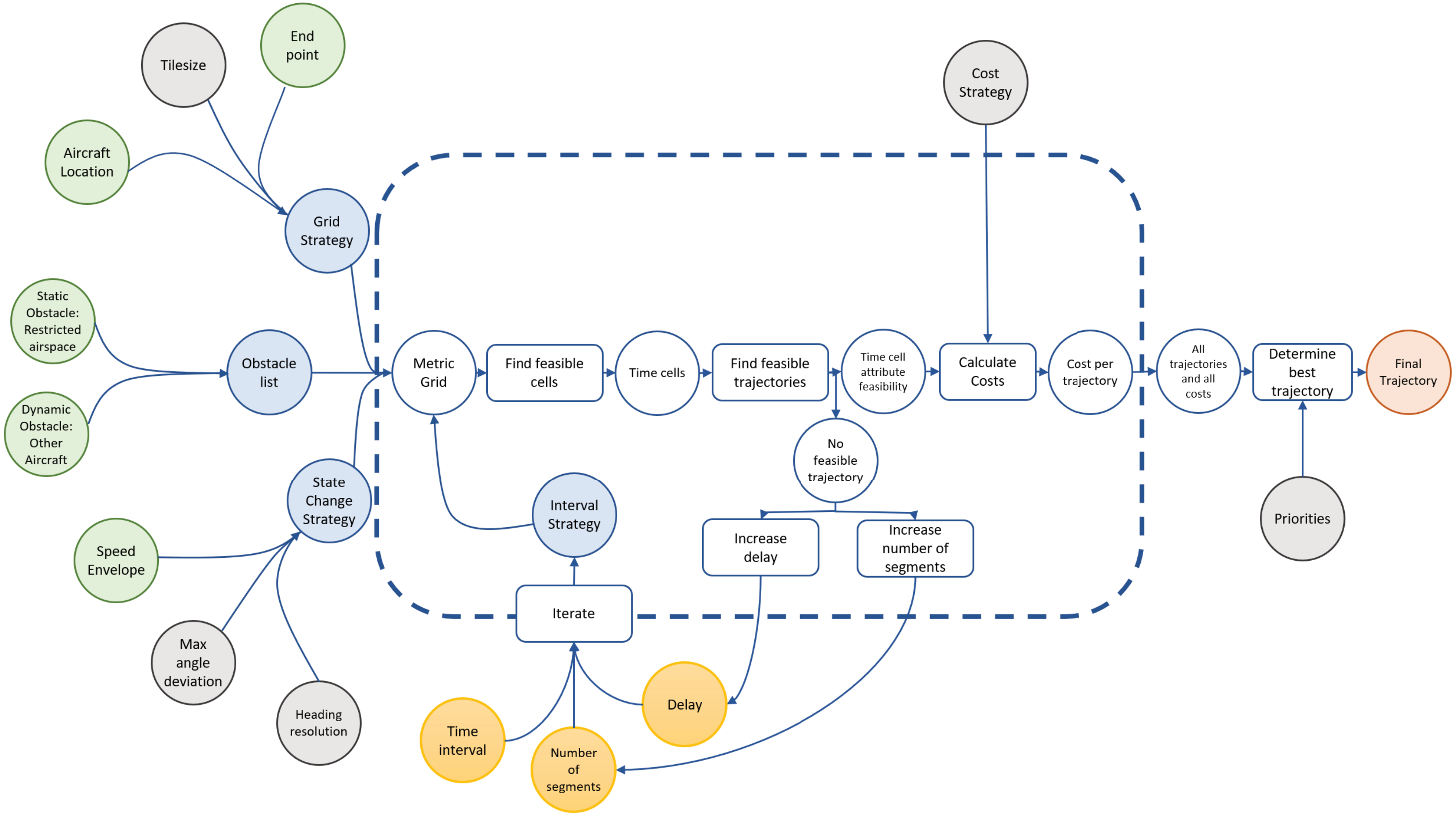


Figure A.12: Overview of algorithm workings. Circles represent information states and blocks are processes. The green circles is information that changes per conflict resolution. The grey circles are constant throughout the whole scenario. The blue circles are the four components that define the metric grid. The blocks within the dashed rectangle will be iterated. Yellow circles are parameters that will be changed throughout the iteration process.

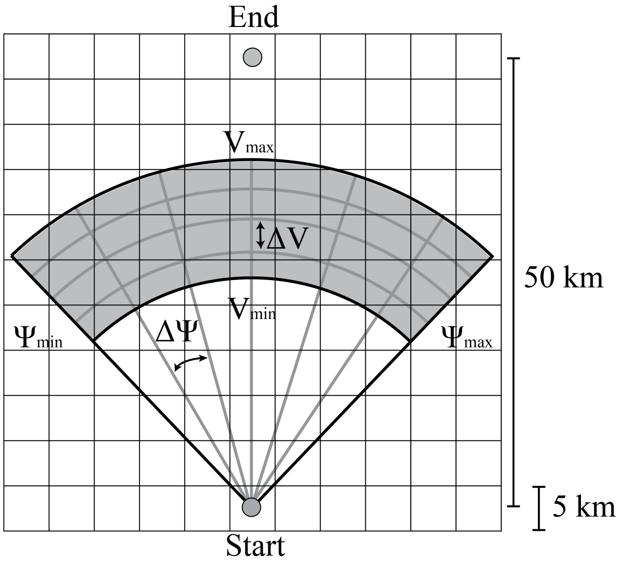


Figure A.13: A grid of 11x11 cells. The possible states for the next time steps are colored grey.

Grid Strategy The grid strategy is composed of a 2D grid in the horizontal plane. It takes the aircraft location, end point location and tile size into account. The grid takes the location of the selected aircraft as a starting point. The end point will be the location of the desired end point. A linear grid space is created between the start and end points. The tile size is a setting that can be changed and still needs to be chosen during display design.

It is possible to use non-linear grids. Using non-linear grids would require extra time to implement, while its contribution to the research is limited. It could be used in order to make the algorithm more efficient or complex. However, multiple other settings, such as tile size, could be adjusted in order to get the same effect. Therefore, using non-linear grids was found to be irrelevant for this research. Furthermore, it is possible to keep the number of grid tiles constant instead of tile size. Using tile sizes is preferred, because the resolution of the grid will remain the same independent of the distance between the start and end point.

An example of a 2D grid can be seen in Figure A.13 The tile size that is used in the example is 5km. As the distance between start and end points is 50km, the grid has 11x11 tiles. Note that the algorithm considers the center of the tile as the waypoint for further evaluation.

Interval Strategy The next part of the metric grid is the interval strategy. The interval strategy adds the dimenision of time. It creates empty time cell maps at time intervals. A time cell map is a structure that maps time cells to a particular time. Time cells are datapoints that are created during forward propagation and contain important information about the feasibility, branch possibilities and adaptability/robustness costs of the time cell itself.

As can be seen in Figure A.12, the interval strategy takes three parameters: number of segments, time interval and delay. In this research, the algorithm is limited to placing only one additional waypoint during conflict resolution. Therefore, the number of segments is either one or two. This is done in order to keep the workload of the ATCO low. Monitoring multiple waypoints and adjusting them in case automation would recommend an unwanted solution, only contributes to increasing the ATCO workload. Furthermore, re-routing of aircraft with multiple waypoints would not be ideal for pilots and passengers comfort.

The time interval determines at which times a time cell map is created. Figure A.14 shows the creation of time cell maps in case there are two segments. The interval consists of a start time, intermediate time and an end time. The intermediate and end time are represented in Equation A.1 and A.2 respectively. In these equations, P is a percentage of the time at which the algorithm evaluates the placement of a waypoint. Furthermore, the Required Time of Arrival (RTA) is the original end time if no delay is taken into account. The intermediate time is removed in case only one segment is used.

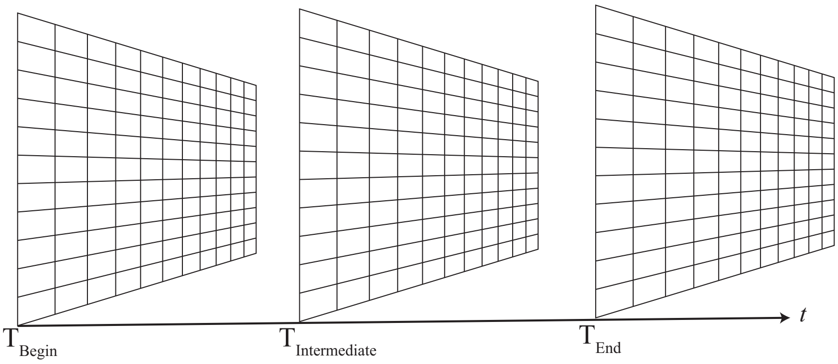


Figure A.14: Time cell maps are created at the time intervals. Each time cell map will have the same structure and are empty still.

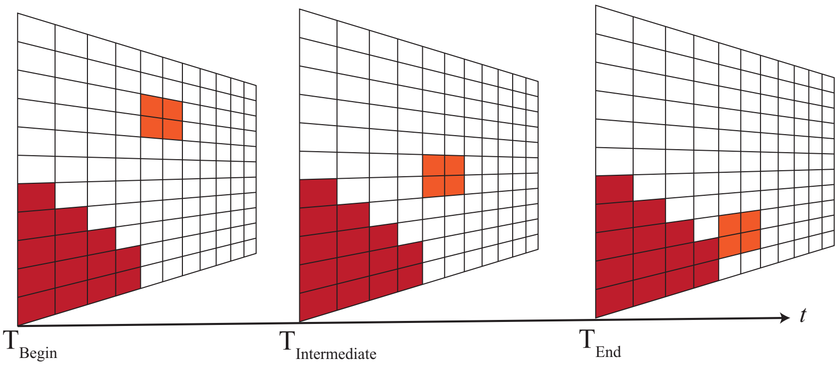


Figure A.15: The time cells that will be blocked due to static obstacles (dark red) or dynamic obstacles (orange).

As is shown in the figure, the delay of the aircraft is integrated into the time interval.

$$T_{intermediate} = P \cdot (RTA + Delay) \tag{A.1}$$

$$T_{end} = RTA + Delay \tag{A.2}$$

Obstacle List The obstacle list is the third part of the metric grid. The obstacle list has two types of obstacles: dynamic and static. Static obstacles block the same time cells and branches for all time steps. Dynamic obstacles move through the airspace and the time cells that they block will be different for every time step. A restricted airspace sector is an example of a static obstacle. Dynamic obstacles could be travelling thunderstorms or other aircraft in the sector. The obstacle list is used to identify potential conflicts in between time steps by extrapolating their tracks. In Figure A.15, the static and dynamic obstacles are visualised. The dark red coloured cells resemble the time cells that will be blocked due to static obstacles. The orange coloured ones match the dynamic obstacles.

State Change Strategy The final part of the metric grid is the state change strategy. It defines how the aircraft is able to change its state for the next time step. It consists of a speed and a heading strategy. The speed strategy is defined by the minimum speed, V_{min}, and the maximum speed, V_{max}, of the aircraft. Furthermore, a step size, ΔV, is defined. The heading strategy is built up similarly. A minimum, Ψ_{min}, and maximum heading, Ψ_{max}, are defined together with a step size of possible change in heading, $\Delta\Psi$. The algorithm assumes instantaneous heading and speed changes. An example of a state change strategy on a 2D grid can be seen in Figure A.13.

A.3.1.2 Forward propagation: finding all reachable cells

The next step in the algorithm is forward propagation. During this step, the reachable cells are explored. The first time cell map is the start. The state change strategy determines which cells in the next time

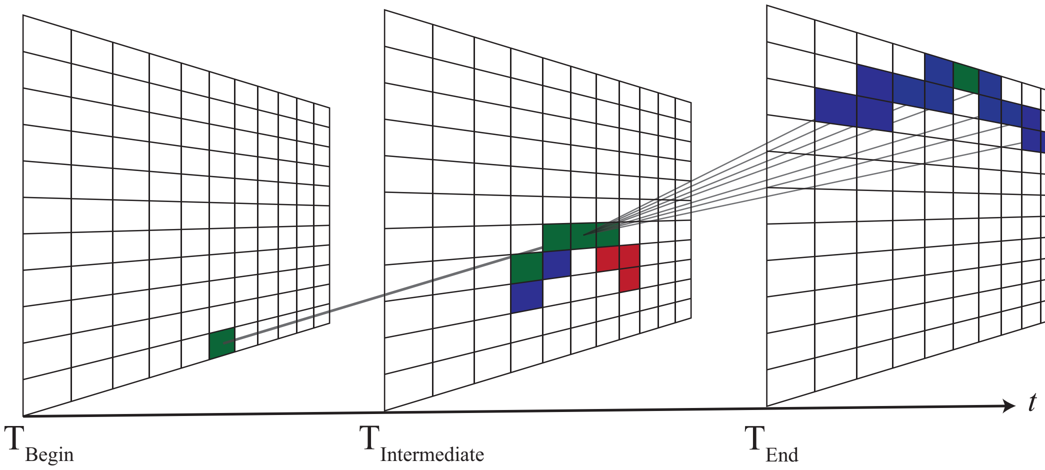


Figure A.16: Forward and backward propagation: The red cells are time cell that have not been created due to an obstacle. The blue cells are cells that the aircraft could reach, however starting from those time cells the end point is not reachable. The green cells are both reachable from the start point of the previous time cell map and are able to reach the end point of the next.

step will be evaluated. It is checked if the cells are blocked by an obstacle. This could be the case if the time cell is reserved for an obstacle. It is also possible that a branch to the cell is intersected by a static or dynamic obstacle. If no obstruction is found, a time cell is created. All time cells are thus reachable states as they fall within the state change strategy and are not blocked due to an obstacle. For all the new time cells in the time cell map, the process will start again for the next time cell map.

Figure A.16 shows the forward propagation. All cells that are evaluated have a colour. If no time cell could be created due to obstacles, the cell is coloured red. A time cell is indicated with either blue or green. The difference between these two will be detailed further ahead. For simplification, the last time cell map shows only the time cells of the upper time cell of the second time cell map in the middle of the grid.

A.3.1.3 Backward propagation: finding all feasible trajectories

During backward propagation, the time cell maps are analysed in a reverse order in order to find the feasible trajectories. During this process the end point will be marked as feasible. Then it will be checked which time cells have feasible branches to the end cell. These time cells are marked feasible as well. This process repeats until it reaches the start point. All cells that are now marked as feasible are not only reachable, it is also possible to construct a trajectory from start to end using that cell.

Figure A.16, shows the backward propagation as well. The green coloured cells are marked as feasible. The blue time cells are not feasible. The backward propagation starts from the end point in the last time cell map. Therefore, only one time cell is coloured green in this map. All cells that are connected to this cell via a feasible branch are coloured green as well.

A.3.1.4 Finding the optimal trajectory

The final task for the algorithm is selecting the optimal trajectory out of all feasible trajectories. This is done using a cost function. In total, there are 4 implemented cost functions and these are the same as used by Ten Brink [71], which are based upon the work of Idris et al. [37]. These cost functions optimize for shortest path, robustness, adaptability or a combination of them. The concepts of robustness and adaptability are defined as:

> “**Robustness** is defined as the ability of the aircraft to keep its planned trajectory unchanged in response to the occurrence of disturbances, for example, no matter which trajectory or conflict instances materialize.” [36, p. 4]

> “**Adaptability** is defined as the ability of the aircraft to change its planned trajectory in response to the occurrence of a disturbance that renders the current planned trajectory infeasible.” [36, p. 4]

Ten Brink [71] found in his preliminary research that the results of using adaptability and robustness would only differ if six segments or more were used. During this research, the maximum number of segments is two. Therefore, adaptability is not considered in this research. Furthermore, during the designing stage, the focus is on the shortest path strategy. Only if time allows it, other strategies will be used. The cost functions related to the concepts stated above, are defined as follows:

$$Shortest\ path = SP = Minimize(Trajectory\ length) \tag{A.3}$$

$$Adaptability = ADP = Maximize(Number\ of\ reachable\ trajectories) \tag{A.4}$$

$$Robustness = RBT = Maximize(\frac{Number\ of\ feasible\ trajectories}{Number\ of\ reachable\ trajectories}) \tag{A.5}$$

$$Combination = Maximize(-C_{SP} \cdot SP + C_{ADP} \cdot ADP + C_{RBT} \cdot RBT) \tag{A.6}$$

C_{SP}, C_{ADP}, C_{ROB} are the weights that are used to combine the costs of shortest path, adaptability and robustness respectively. These still need to be determined.

A.3.2. Verification

Ten Brink [71], verified the algorithm using the TSR interface. Verification of the code needs to be done again, because some minor adjustments were made and the code has been integrated in new simulation software. The idea is to verify it using the TSR again. The algorithm creates trajectories by placing intermediate waypoints. This is similar to the approach of the TSR interface. This verification will be done at the beginning of the next research phase.

A.3.3. Limitations

The Flexibility Metric algorithm has some limitations itself. This section touches upon some of the limitations that are relevant to the research at hand. The first limitation that becomes clear from the description of the algorithm workings is that the algorithm discretizes the work domain. This is done both in space and time. This creates the possibility that some theoretical solutions will be overlooked due to the discretisation. Fortunately, the chosen discretization in space can be made using small steps largely solving the problem. Primarily for discretization in time, computational expensiveness will be influenced more by minimizing the time steps. The right balance needs to be found between computational effort and solution resolution.

Furthermore, changes in speed and heading are assumed to be instantaneous. The software program in which the algorithm is integrated, takes into account flight dynamics. Therefore, problems may occur for conflicts at a very short time to collision, because the simulated aircraft is not able to react as fast as the algorithm thinks the aircraft is able to. Therefore, it was chosen to only evaluate conflicts with over 5min to LOS.

Finally, the algorithm needs to function suboptimally in order for the operator to understand what is going on. The suboptimality might be induced by increasing the tile size or time steps in order to make them visible. Additionally, discretizations in the state change strategy need to be made. The algorithm will only evaluate at some cells in a certain heading and speed range. More simplifications might be required in order to be able to increase the internal transparency of the algorithm.

A.3.4. Insights for explanations

In the final section of this chapter, the information that is relevant for constructing explanations regarding the algorithms workings is summarized. In order to show how the algorithm is interpreting the work domain, the four elements that build the metric grid are of importance. These are the grid strategy, interval strategy, obstacle list and state change strategy. This will be a starting point for creating a higher level transparency.

Other relevant information could be the cells that the algorithm is evaluating. Additionally, showing time cells that are reachable and/or feasible will be useful. The basis for this information lies in the construction of the metric grid. However, presenting this information could help the operator to show what follows from the defined metric grid.

Other points of interest are the cost strategy and cost values. These determine which trajectories will be chosen. Showing information about the cost strategy and cost values will allow for explanations to be made about why the algorithm prefers some feasible trajectories above others.

Not all information stated above is easily visualised. A multitude of time cell maps will be created and across time cell maps, the same cells will be evaluated. This will lead to different information regarding cost values for the same cells. Summing of time cell maps is therefore not the straightforward solution, but might be an option under some conditions. Choosing the shortest path strategy and limiting the amount of segments and time steps will contribute to the possibility of adding time cell maps.

A.4. Automation Transparency

This chapter dives into the concept of automation transparency. Section A.4.1 gives a brief overview of what transparency is. Section A.4.2 explores the benefits and limitations that transparency might have. Transparency will take the form of an explanation in this research. Therefore, Section A.4.3 describes what explanations are and how they are triggered. The techniques that could be used for explaining algorithms are explored in Section A.4.4. In addition, the relevance of those techniques for the design of the interface are discussed. Finally, Section A.4.5 will conclude with the finding of this chapter and will present key steps for further analysis for constructing explanations and increasing the transparency of the algorithm.

A.4.1. What is transparency?

The word transparency has many meanings and is used across many fields of research. In this research, transparency regarding automation by algorithms is considered. Normally this type of transparency is referred to as *algorithmic transparency*, *automation transparency* or *agent transparency*. The following definition for transparency will be used [79, p. 2]: “The extent to which aspects of the automation’s inner process underlying a solution can be observed and explained in human terms.” Well implemented transparency could enable the operator to get insights of the inner workings of automation, which will allow for answering questions about why and how the computer agent makes decisions. Transparency will thus take the form of an explanation [70].

The opposite of transparency is opacity. Opacity in algorithms refers to the occluding of the inner workings of the algorithm to its user [58]. In literature, both terms are used and essentially tackle the same problem. Transparency is usually presented as the solution for opacity. Another well known formulation for opacity is the black box model. Inputs and outputs are known, however the link between in- and output is not clear for the user. Opacity in algorithms can have many downsides such as automation surprises, biases and low trust in automation [20].

According to [52] transparency consists of two elements: *accessibility* and *comprehensibility* of information. The operator must be given (relevant) information about the decision process of automation. In addition, the information should be presented in such a way that it can be (easily) understood by the operator. The combination of accessibility and comprehensibility is interesting because they are in potential conflict. Disclosing more information about the decision process in real time, could endanger comprehensibility by overwhelming the user with information.

The remainder of this section will be dedicated to subdividing transparency to clarify terms used in this report.

Explainability and Auditability A first distinction in transparency that can be made is described with respect to the goals for using transparency in interface design. According to Springer and Whittaker [69], transparency has two facets: *explainability* and *auditability*. Explainability has as a main goal to improve the user experience and usage of the system by showing the user the inner working of the system. The information presented is incomplete, but it is enough to enhance the user mental model and increase user trust. Auditable transparency is mainly used for validating the system for use by external parties. The aim of this type of transparency is to ensure that the system is fair and unbiased. Complete and sound information is required for this purpose.

Springer and Whittaker argue that when designing for transparency, it is not possible to achieve both goals simultaneously with the same implementation. A choice has to be made between the facets. They claim that the user will be overwhelmed with the information required for proving fairness which will not contribute to the usage of the system and to the user experience.

In this research, the focus will be on explainable transparency. Improving trust and performance of the system are the main goals of using transparency. These goals are aligned with the explainable facet of transparency. In general, it is also important that operators are able to detect the limits of automation. However, the interface should give enough comprehensible information to detect these limits, while not overwhelming the operator. Detecting algorithm biases or design flaws should be done by another form of transparency for another audience.

Knowing that the focus will be on explainable transparency, it is worthwhile to differentiate transparency further into “Domain and Agent Transparency” and “Operator and Designer Transparency”. These concepts will be used throughout the report, therefore the difference between these transparencies is described in the paragraphs below.

Domain and Agent Transparency An important distinction can be made between transparency in the work domain and transparency for the inner workings of the algorithm. A good example is the TSR, it shows the constraints based upon the work domain. Due to speed, heading and obstacles there are areas where no waypoints could be placed (red areas) and places where it is possible to place an intermediate waypoint (green areas). The representation is continuous as the constraints from the work domain are continuous. However, if the algorithm is the subject for transparency, continuous constraints may not be applicable. If an algorithm analyses the environment and produces discrete solutions, it might be interesting to show the discretized constraint instead of the continuous variant as it gives more insight into how the algorithm works. It is an important aspect to keep in mind during the design process.

Operator and Designer Transparency Another important aspect for explaining is the target audience. Who is the explainee and what is their base of understanding; their starting point for interpreting the problem? This results in designer and operator transparency. Both have different needs for explaining. The designer might want to know in detail how the algorithm works. The operator might only be interested in why it makes that particular decision and is not interested in the exact workings of it. Furthermore, the operator has only limited time to “be explained” and must not be overwhelmed with information. Clearly the explanation must be tailor-made for the audience. That doesn’t mean that elements of designer explainability could not be used in operator transparency and vice versa. During the design process a clear definition needs to be made on what the audience of the transparency will be.

In this research the goal is to explain how the algorithm makes a decision and why it makes a particular decision. Explainable transparency for the operator that focuses on the inner workings algorithm is thus required. Transparency will take the form of an explanation, which will be visualised either explicitly or implicitly.

A.4.2. Benefits and limitations

The first questions to answer are: why is transparency needed? And what benefits does it gives us? In this section, these questions will be treated in more detail. A brief summary of limitations in transparency will be given as well.

A.4.2.1 Benefits of transparency

Many articles indicate the potential for transparency. Unclear reasoning of automation results in questioning the automation’s accuracy and effectiveness by human operators. Transparency is proposed as a solution by [14], [24] and [29]. In [84], transparency is proposed for adoption of automation in the medical field. A field that is similar to ATC in the sense that it has high stakes and requires high expertise by the operator. In [21] transparency is suggested as a mean to increase the mental model of the user of the system as a whole, which in turn increases trust, control effectiveness and user’s satisfaction according to [17], [45] and [49]. Furthermore, transparency could be used in addition to strategic conformance [82] or could (partially) replace it. It is expected that adding transparency to the system will result in the opportunity to make the system less conformal, which could potentially lead to more optimal solutions [79]. Finally, according to [52] transparency is required for allowing meaningful oversight by the human. They state that human intervention becomes increasingly difficult if the human has less information than the machine. This comes down to a call for transparency to increase trust and understanding as well as control over automation [1].

A.4.2.2 Limitations of transparency

Literature in the field of transparency is not only positive. Critical voices are trying to temper the overall (naive) enthusiasm for transparency [52]. In [4] a total of 10 limitations have been summarized. Five of these limitations were found relevant for this research. The other five limitations are primarily focused on large social-technical systems, such as banks or governmental institutions, in which processes needed to be made transparent to the general public. The five relevant limitations are listed below:

1. **Transparency can (intentionally) occlude**: Due to a large amount of information, users could be overwhelmed and might not see important pieces of information.

2. **Transparency does not necessarily build trust**: Research shows that *full transparency* could have a negative effect on trust [42]. In addition, transparency could give a false suggestion of trust [41].

3. **Transparency can privilege seeing over understanding**: Seeing is not the same as understanding. Interaction is required to understand how systems behave in relation to their environment [62].

4. **Transparency has technical limitations**: Some parts of algorithms might be hard and sometimes impossible to understand. Even for creators, algorithms are sometimes inscrutable [13]. In [18], it is argued that not all relevant information could be disclosed due to the fact that some information never takes an observable form.

5. **Transparency has temporal limitations**: Different moments in time may require or produce different kinds of system transparency.

In [41], two other interesting limitations become clear. It is argued that transparency fundamentally depends on having a critical audience. Without a critical audience, flaws in automation cannot be traced and the potentials of transparency are limited. Furthermore, algorithms become more complex and perform with increasingly greater speed, which is in contrast with humans who are limited to their own cognitive resources [41]. This results in the following two extra limitations:

6. **Transparency needs a critical audience**

7. **Transparency is limited by human cognitive resources**

For this research, it is important to keep these limitations in mind while designing the display.

A.4.3. Explanations

As indicated by the sections above, transparency has to be able to explain the algorithm's inner workings. Explanations are thus a central concept in this research. Two major question arise by formulating explanations: “What to explain?” and “How to explain?” [21]. The following section takes a closer look at explanations and will find clues to answer those questions. Unfortunately, answering these questions is not straightforward and depends on the use-case and the audience.

Table A.2: Table with triggers for explanation and the related user learning goals. The most relevant triggers and user's goal are colored. Retrieved from [33]

TRIGGERS	**USER/LEARNER'S GOAL**
How do I use it?	Achieve the primary ask goals
How does it work?	Feeling of satisfaction at having achieved an understanding of the system, in general (global understanding)
What did it just do?	Feeling of satisfaction at having achieved a understanding of how the system made a particular decision (local understanding)
What does it achieve?	Understanding of the system's functions and uses
What will it do next?	Feeling of trust based on the observability and predictability of the system
How much effort will this take?	Feeling of effectiveness and achievement of the primary task goals
What do I do if it gets it wrong?	Desire to avoid mistakes
How do I avoid the failure modes?	Desire to mitigate errors
What would it have done if x were different?	Resolution of curiosity at having achieved an understanding of the system
Why didn't it do z?	Resolution of curiosity at having achieved an understanding of the local decision

A.4.3.1 Triggers for explanations

There are many triggers that could result in the request for an explanation. Table A.2 gives an overview of potential triggers [33]. Not all triggers are relevant for this research. The triggers are dependent upon the situation and the audience. One can imagine that a novice might have different learning goals compared to experts. The same holds for the designer of the algorithm or interface compared to the ATCO.

The scope for this research has been narrowed down to operators working in an Area Control Centre. They have experienced in the ATC tasks, which are experts in their field, however they are not necessarily experts in the algorithm. This gives clues to which trigger are of importance in this research and which triggers are not.

Triggers that have been found out of scope are "How do I use it?", "How does it work?", "What does it achieve?", "How much effort will this take?" and "What do I do if it gets it wrong?". These questions are relevant for the training phase of novice users. They help the user learn the system. Explanation coupled to these question could be of use for the introduction of an explainable system. However, during operation, these explanations could be considered information noise, because the operator already knows the answers to those questions.

The question "How do I avoid the failure mode?" is trickier. The operator does not have influence on an automation failure. However, he should be able to notice it. In addition, showing the limitations of the automation is interesting as well. This will give the operator the opportunity to judge the quality of the proposed solution.

The questions "What did it just do?", "What would it have done if x were different?" and "Why didn't it do z?" are the most relevant triggers for explanation within the scope of this research. These questions are focused on the local understanding of a decision rather than the global understanding of the system. Local understanding is defined as the understanding of a specific decision or single prediction. Global understanding is defined as understanding of the whole logic of the model. Using that knowledge the operator should then be able to explain all different possible outcomes. These definitions are based upon the definition of local and global interpretability [2] and local explanations [6]. Local understanding thus deals with the question: "why has *this* decision been chosen in *this* situation?". Local understanding solves anomalies that an operator might have and will help him to judge the current decision. The focus is thus on the current situation making local understanding more desirable during operation.

Even though the focus will be on local understanding, some degree of global understanding must be present. Global understanding is able to facilitate local understanding. Furthermore, in novel situations global understanding might be required to help the operator in Knowledge Based Behaviour (KBB) (e.g., determining whether the algorithm is capable of solving that problem considering the limitations that it has). It should be noted that the global understanding should mainly be created during the training phase. During operation, information presented by the interface will facilitate the use of global understanding on the particular problem.

The explanations that give answer to the above mentioned questions could be given explicitly or implicitly. The difference between these are that explicit explanations directly compose an answer to the question, e.g.: “This waypoint was not chosen, because it has worse performance regarding distance”. Implicit explanation will show the required information to interpret the reason for a certain decision. The required information could be aided with clues of importance in order to help interpretation.

A.4.3.2 Human explanations

An important aspect of making the automation transparent is the human itself. As is indicated in Section A.4.2.2, one of the limitations of transparency is introduced by the limited cognitive resources. The human should be able to process the amount of information and in addition the information should be comprehensible as well. In general, the human factor has been taken into account insufficiently in explainable AI [1].

Miller [51] connects the research in explainable AI with the vast amount of research on explanations from the social sciences. The main conclusion is that explanations are *contextual*. This conclusion is based upon the following major findings:

- **Explanations are contrastive:** People do not ask why event X happened, but are more interested in why event X happened instead of event Y. Explanations are needed in response to a counterfactual case, also known as foils.
- **Explanations are selected (in a biased manner):** People are generally not interested in all the causes for event X to happen. They are inclined to take only one or two causes to explain event X. These primary causes are usually chosen in a biased manner.
- **Probabilities do not matter:** Referring to probabilities and statistical relationships is not as effective as referring to causes.
- **Explanations are social:** Explanations are transfers of knowledge in the form of a conversation or interaction.

The findings of Miller should be used as guidelines in how explanation could be best presented to humans. During the design of the interface, these main conclusion should be taken into account. Note that probabilities can still be valuable, however causes have the preference if they are available.

A.4.3.3 Progressive disclosure

As discussed earlier in this section, there are triggers that will result in the request of an explanation. This gives an indication of what to explain. However, it does not tell us when an explanation is needed. Explanations containing too much information could be harmful to the usage, trust and performance of the system [42]. Furthermore, the excessive information could overwhelm the user leading to rejecting the tool according to Springer et al. [70]. Explanations are only needed if expectation are violated.

Springer et al. propose the usage of progressive disclosure. This means that information is given on an “as needed” basis. This is similar to how humans usually provide explanations as this is done when the situation demands it [26]. By using progressive disclosure, the user will dive deeper and deeper into the algorithm on request. It will help select what information is relevant for the current situation. This way progressive disclosure could help avoid display clutter, while maintaining the ability to give complete information if necessary. In order to create an interface that incorporates the idea of progressive disclosure, interaction will be of importance.

The idea of progressive disclosure could be used together with the concepts of SBB, RBB and KBB as described by the SRK taxonomy [61]. This taxonomy will be explained in more detail in Section A.5.5. Information for RBB during an uncommon situation, might be stored at a deeper level than information for commonly used SBB and RBB. Another deeper level could store information required for KBB, which is usually only used in new and unfamiliar situations.

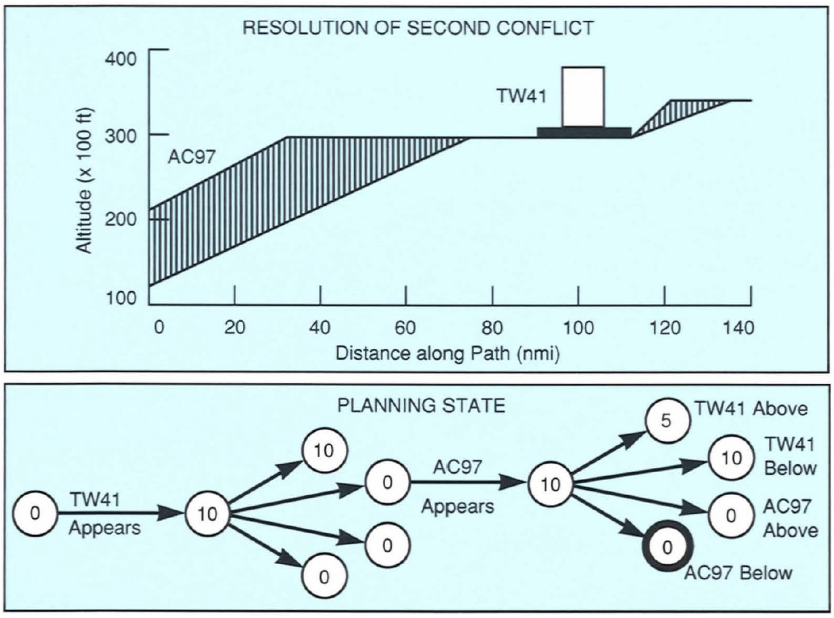


Figure A.17: Decision tree example. Upper figure resembles traffic situation and lower figure a decision tree explaining algorithm decision for aircraft planning [67].

A.4.4. Techniques for explaining algorithms

The TSR and SSD are used as inspiration and will form a basis that is able to show the constraints applied by the work domain. In order to create an interface that explains the Flexibility Metric algorithm's inner-workings, it is interesting to look into explanation techniques. This section will show a variety of techniques that could help explaining algorithms. These techniques do not necessarily have to be used separately.

A.4.4.1 Decision trees and tables

One commonly used technique for explaining algorithm rationale is decision trees [2]. Huysmans et al. [34] present research into the comprehensibility of decision trees and tables. It was found that larger and more complex representations resulted in a decrease in answer accuracy, an increase in answer time and a decrease in confidence. Decision trees, especially, were found to have decreased performance due to their rapidly expanding representation size for large problems. Decision trees and tables should thus only be used for representations of low complexity.

Spencer [67] used decision trees to visualize the planning state of his newly developed algorithm for ATC automation. In Figure A.17, the top figure represents the traffic situation, while the bottom figure resembles the planning state. The algorithm is always selecting the lowest score possible in planning the action for the aircraft in question. The simple representation clearly tells the operator what consideration the algorithm has made and why the algorithm has made a decision.

Decision trees are model specific and their effectiveness and ease of implementation is therefore also very dependent upon the algorithm under consideration [2]. The Flexibility Metric algorithm, used in this research, does not necessarily lend itself to a decision tree. The algorithm will evaluate a large number of possible trajectories. Displaying all trajectories in the form of a decision tree will consume a large part of the screen, if it fits at all. Furthermore, large decision trees require great effort for processing and interpreting. A surrogate model or a trajectory selection module should be developed if decision trees were to be applied for explaining purposes.

A.4.4.2 Contrastive Explanations

In Section A.4.3.2, it is stated that humans often create explanation in a contrastive manner. The implicit question is usually “Why did event X happen instead of event Y?”. Therefore it seems logical to build explanation of algorithms in a similar way; creating contrastive explanations. These explanations could take multiple forms. In this literature study both counterfactuals and “What if” and “Why not?” explanations are considered.

Counterfactuals Adadi et al. [2] define counterfactual explanations as explanations that show the minimum condition that would have led to an alternative solution. The counterfactuals are the conditions that will change the solution. An example would be: “The solution would change if x>a”. “x>a” would be the counterfactual in this example. The idea might also be visualised. An example would be colouring the area or values that would ensure the same solution.

This explanation contributes to creating local understanding as it gives information about why *this* certain decision was chosen and why *this* decision is no longer true. According to Grover et al. [27], the explanatory power would be even greater in accompanying counterfactual with factual reasoning; so called “balanced” explanations.

Building counterfactual explanations might not be straightforward. Guidotti [28] is using a genetic algorithm to learn the local decision boundary in the neighboorhood of the instance under consideration. A decision tree is created using the logic found. The decision tree is then used to build factual and counterfactual explanations. This process of developing an explanation is extensive and might not be the way to go for real-time operations. At least some degree of logic has to be extracted from the algorithm used in this research in order to create counterfactual explanations.

“Why not?” and “What if?” Another way to create contrastive explanations is by facilitating answers to “What if?” and “Why not?” questions. Both questions allow for exploring the field of possibilities. The difference between the question is in the nature of the trigger as discussed in Section A.2. The user’s learning goal for “What if” questions is curiosity about how the system will work. “Why not questions” are more focused on answering the anomaly a user would have considering a local decision [47]. “Why not” questions will thus require a higher degree of explanatory information.

Lim et al. [47] research “What if” and “Why not” explanations. They found out that ”Why not” explanations significantly increase understanding, trust in the system and task performance. As the task in their research was focused on interpreting and evaluating, “What if” explanations merely performed better than no explanation at all. Other tasks could still benefit from “What if” explanations.

A way of implementing “What if” and “Why not” explanations would be to let the user select a point of interest for placing a waypoint. The interface should then give performance results of the trajectory related to that point of interest. This implementation still has limitations as it is only able to show some trajectories and not all. Therefore, the user needs to actively probe options.

A.4.4.3 Interaction

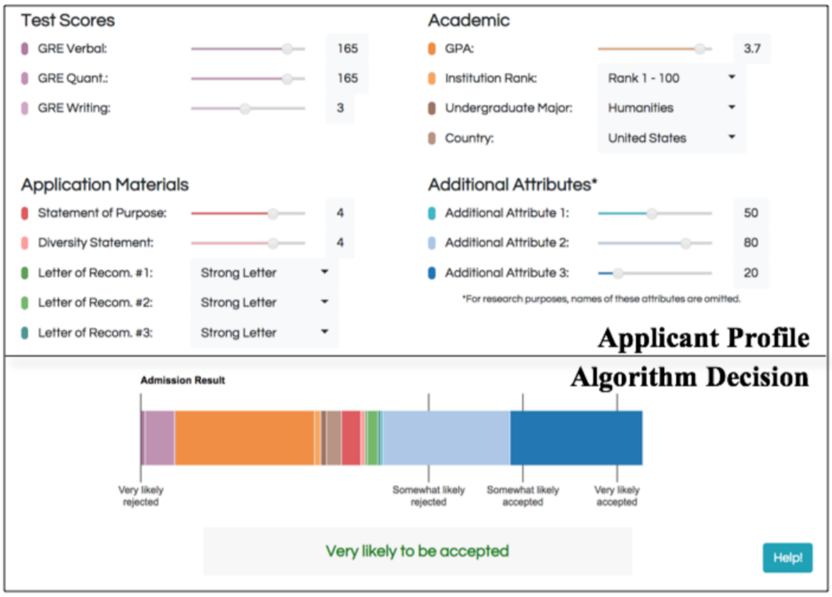
“Explanations are social” is one of the concluding remarks from Miller’s review on explanations. It states that explaining happens in a dialog between explainer and explainnee: a form of interaction. As of now, explanations resulting from research of the Explainable AI community are static [1]. Abdul et al. suggest an alternative approach in which users are allowed to explore the system’s behaviour freely through interactive explanations.

Interaction is closely related to progressive disclosure described in Section A.4.3.3. Interactively giving more information at user request only if the user is in need of that information. This will ensure more effective explanations and helps in determining the foil of the explanation requested by the user.

Apart from the concept of progressive disclosure, this technique could be used in combination with many other techniques. Another example is the combination of interaction with partial dependence. Interaction could be in the form of sliding bars, that would change the input or settings of the algorithm. The operator could then see what the effect is of a particular input or setting on the output.

An example of such an interface can be seen in Figure A.18. The user is able to explore the effects of inputs, e.g. “test scores”, on the algorithm decision. In addition, it is visualised how much each value contributes to the final decision. Cheng et al. [15] used this interface in their research for finding design principles for explanation interfaces. They found that interactive explanations can improve understanding. However, interactive explanations take time when compared to static ’white box’ explanations. Note that their research had a target audience of non-experts.

It should be noted that there should be a balance between the interactivity and stasis of the explanations. If the operator needs to perform too many steps, it will require a lot of effort to find the right


Test Scores

GRE Verbal: 165

GRE Quant.: 165

GRE Writing: 3

Academic

GPA: 3.7

Institution Rank: Rank 1 - 100

Undergraduate Major: Humanities

Country: United States

Application Materials

Statement of Purpose: 4

Diversity Statement: 4

Letter of Recom. #1: Strong Letter

Letter of Recom. #2: Strong Letter

Letter of Recom. #3: Strong Letter

Additional Attributes*

Additional Attribute 1: 50

Additional Attribute 2: 80

Additional Attribute 3: 20

*For research purposes, names of these attributes are omitted.

Very likely to be accepted

Figure A.18: Sliding bars as an interactive element for explaining a student admission algorithm [15]

explanation for the anomaly the operator wants to solve. At the same time, static explanation might not allow for efficient selecting of information and might even suffer from information incompleteness due to display constraints.

A.4.4.4 Animations

Animations are used for learning environments and are able to increase understanding. Vegh and Stoffova [76] showed that their card interactive animations helped students understand a sorting algorithm. The students were able to recognize the main ideas of the sorting algorithm, but they did not understand the algorithms in detail. Vegh and Stoffova claim that more specific micro level explanation should be used to also explain the details.

This highlights the challenge of using animation for explainability purposes. It should be carefully considered what needs to be explained. The design challenge is to skip over small steps in order to visualize the major step or focus on micro-steps at the expense of not visualizing the major step [53].

Animation could be used together with the concept of slowness. Algorithms are making decisions much faster than humans. Slowing down the algorithm reasoning process and visualizing the reasoning step by step could help humans further understand the algorithm. Furthermore, according to Park et al. [57], slowing down algorithms would encourage users to reflect on judgements of themselves and the algorithm. Results showed that participants accepted good quality algorithms more and bad quality algorithms somewhat less due to slowness.

Animation has been proven to mainly increase understanding under learning conditions (e.g., [76]). Animation might thus be very suitable for operators that are novices. Using animation, the main ideas of the algorithm workings could be taught to the operator. However, experts request explanations for solving anomalies rather than to learn the main idea behind the algorithm. Furthermore, one could raise questions at the relevance of showing the experienced operator an animation that he has already seen a dozen of times. Animations are therefore found to be less relevant for this research. The concept of slowness could still be of interest as it will allow operators to reflect on algorithm decisions.

A.4.4.5 Partial Relationship and Sensitivity

Another way of explaining what happens inside automation, is by using partial relationship. Visualising the partial relationship between an input variable and the outcome of automation will tell the user something about the inner working of automation [2].

As discussed in Section A.4.4.3, sliding bars could be used to visualize this relationship. Changing the input will have some effect on the overall output (or none). This information tells the user something about the most important input variables and the sensitivity of the solution regarding these input variables. This information could be used to judge the validity of the proposed solution.

Another example is shown in Figure A.19. This example makes use of Shapley explanations to create understanding of the algorithm [48]. The purple features are contributing to the risk of getting a disease, while the green values are decreasing the risk. The explanation clearly shows how the outcome is built up. These Shapley explanations are especially interesting when multiple input variables influence the outcome.

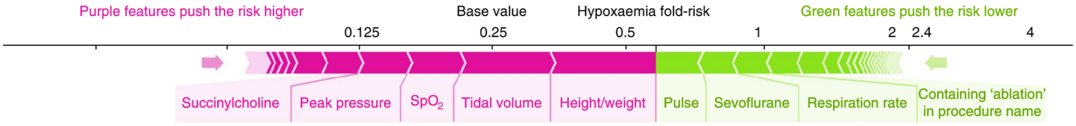


Figure A.19: Shapely explanations in the medical field [48]

A final example is the SUPEROPT interface developed by Richards and Turnbull [72]. This interface will allow for controlling the cost function by imposing constraints to which the algorithm needs to adhere. The SUPEROPT interface has a different goal than this research. Its goal is to control and interact with the algorithm. However, the idea of influencing the outcome of the algorithm by changing the settings and strategies could be used for an increased understanding using partial relationships.

If these types of explanations were to be implemented, the number of inputs to be checked should be limited as it is not ideal for the operator to look at the partial relationship of every input or setting separately. Furthermore, the main goal has to remain understanding of the algorithm and not the interactions with and modification of algorithms proposed resolutions.

A.4.5. Conclusion

From the section above, it could be concluded that transparency should improve explainability of the system by showing the inner workings of the algorithm. Transparency can have many benefits. However, transparency is not the answer to all problems and has a number of limitations. It is important that the audience stays critical and that the human limitations of its cognitive resources are considered.

Therefore, the human way of providing explanation should be taken into account during the design. Two major conclusions of Miller [51] are that explanations are contrastive and social. Therefore, “what if?” and “why not?” explanations will be of interest during the design. Furthermore, the aim should be to make use of interaction as well. Partial relationship in the form of Shapley explanations or sliding bars, could be used if a combination strategy will be used or multiple inputs/settings influence the outcome greatly.

Answering the questions of “What to explain?” and “How to explain?” is very dependent upon the use-case and the audience itself [21]. In this chapter, a start has been made at answering these question for the use-case and audience of this research. The answers are still generic and could be more specific. Cognitive work analysis, Chapter A.5, and an analysis of the Flexibility Metric, Chapter A.3, could help out in determining what to explain and how to explain.

A.5. Cognitive Work Analysis

The cognitive work analysis is useful for giving designer insights into how to create tools that support human work [77]. It is considered as the basis of ecological interface design and will help discover what needs to be explained. In addition, the CWA puts boundaries on what the user and automation needs to know. According to Vicente [77] a cognitive work analysis consists of five parts: work domain analysis, control task analysis, strategy analysis, social organisation analysis and worker competence analysis. These five steps will all be described in the following sections.

A.5.1. Work Domain Analysis

The first task in the WDA is to scope the work domain and make the system boundaries explicit. The work domain under consideration is an airspace, which is crossed by aircraft that need to travel through the airspace safely and efficiently. Normally, the work domain is independent of the actor that is working in it. However, as we are trying to show the inner-workings of the algorithm that has its own view

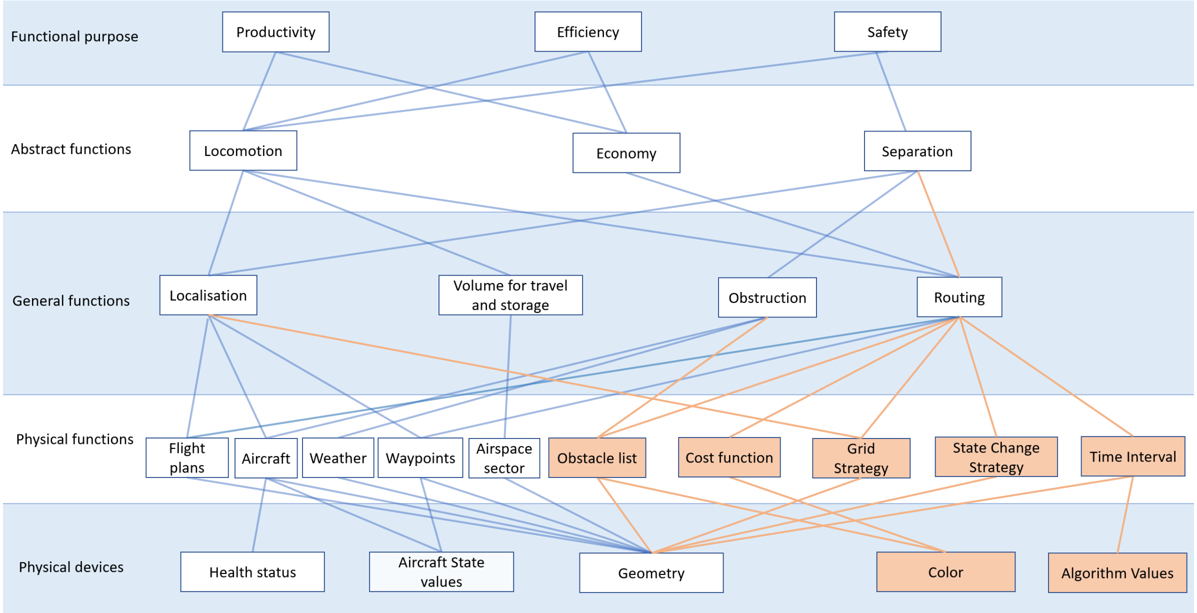


Figure A.20: AH of the work domain based upon AH from Area Control [11]. Orange highlighted blocks are added due to visualisation of the algorithm.

on the work domain, it was chosen to incorporate the algorithm itself into the work domain. This will have influences on the Abstraction Hierarchy. The work domain is further simplified by excluding communication.

A commonly used tool for mapping the work domain is the Abstraction Hierarchy (AH). In Figure A.20, the AH of the work domain is visualised. The AH was based upon a typical Area Control AH retrieved from [11]. Communication has been removed as it is not considered as part of the work domain during this research. The orange blocks and lines are of importance. These have been added as the result of incorporating the algorithm into the work domain.

The algorithm will provide trajectory advisories. Therefore, the algorithm has major influence on the routing block in the generalised functions level. The proposed route of automation is dependent upon the 4 elements that define the metric grid and the cost strategy. The obstacle list, grid strategy, state change strategy and the interval strategy determine how the algorithms sees the work domain. Additionally, these elements of the metric grid will determine which trajectories will be marked as feasible. The cost strategy is used to find a optimal solution from all feasible trajectories. Visualising these 5 elements will show an operator how (re)routing is performed in the work domain.

Other relevant links are connecting localisation and grid strategy, obstruction and obstacles and routing and separation. The orange link between routing and separation has been created, because the algorithm will make sure that aircraft separation is maintained in creating new routing suggestions. Furthermore, localisation and grid strategy are connected due to the major influence of the localisation of the grid strategy. Finally, the link between obstruction and obstacle list has been created. The obstacle list is how the algorithm will interpret obstruction in the airspace.

A.5.2. Control Tasks Analysis

Because this research is focused on supervising an algorithm, the supervisory task of judging an automated conflict resolution has been chosen as the main task to consider. The setting is as follows. An aircraft enters the airspace. Upon entering the airspace, automation will check if re-routing is required. If so, it will determine the optimal trajectory and proposes its resolution. The supervisory task is to check the proposed resolution based on its validity and optimality. The latter being of less importance for the shortest path strategy as human experts usually try to decrease workload by increasing robustness rather than to optimize for the shortest path [44].

Vicente [77] proposes to use a decision ladder during this step of the analysis. Figure A.21 shows the analysis of this task using a decision ladder. The colored blocks and circles are used during the supervisory task. All steps have been noted down in Table A.3. The control task analysis should still

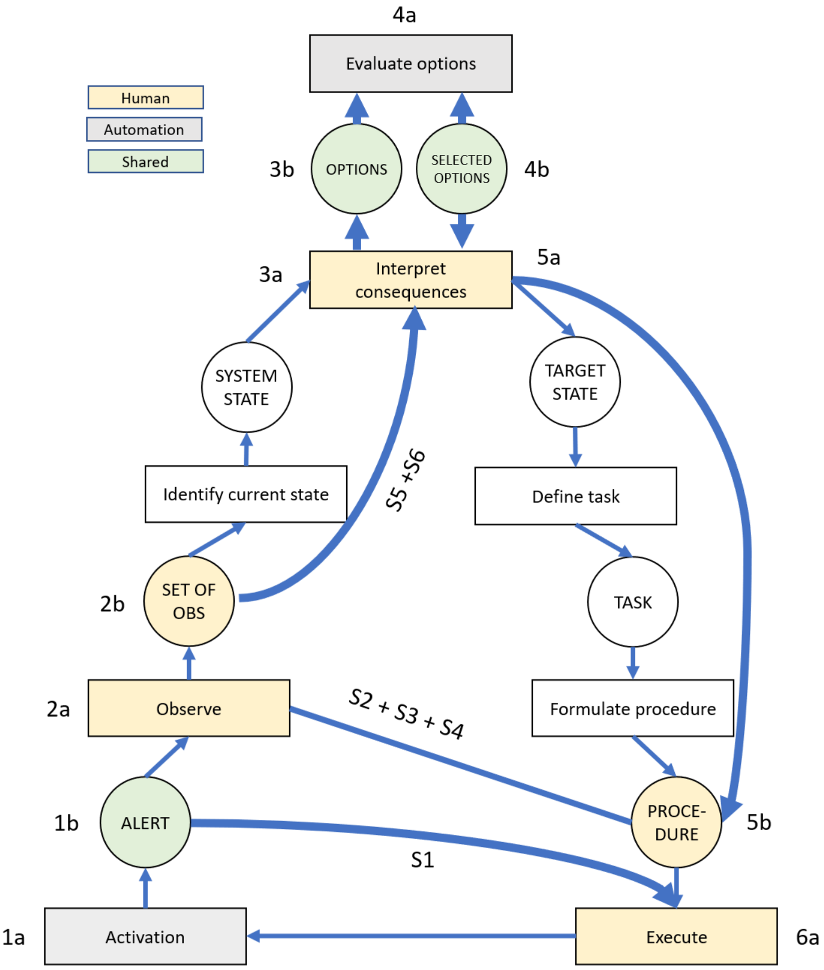


Figure A.21: Control task analysis for supervising an automated conflict resolution using a decision ladder.

be actor-independent. Therefore, all tasks have been described in a actor independent manner. The steps are as follows.

Step 1a is the activation of the task. In this case, activation will be done by the automation's proposed re-routing. The resultant state of knowledge of *step 1b* will be the awareness of the need for supervising the generated trajectory. *Step 2a* is observing the proposed trajectory for re-routing. In *step 2b*, the actor will have knowledge about what the proposed trajectory is in terms of waypoints, heading and speed clearances. The actor will also have knowledge about the performance, costs and safety, of the proposed trajectory. Using this knowledge, a determination is made on whether to go into more extensive investigation of the proposed solution. If so, other potential solutions to the re-routing problem need to be found in *step 3a* and subsequently investigated based on their performance in *step 4a*. All options need to be evaluated in comparison with the known solutions in *step 5a*. The result of this step is knowledge of whether to keep the proposed re-routing solution in *step 5b*. The final step is the execution of the obtained knowledge in *step 6a*.

A.5.3. Strategy Analysis

In the next step of the CWA, the strategies are analysed. During this phase, the goal is to find out how a task could be conducted. Figure A.22 is a strategy analysis flow diagram of the supervisory control task. In this diagram, a start state and an end state form the beginning and end of the diagram. These are “proposed procedure” and “accepted (or rejected) procedure”. 6 strategies have been identified: S1 through S6. These strategies have also been shown in Figure A.21 as shortcuts.

S1 is a simple strategy in which the actor does not use any information in determining whether to accept the decision or not. It will just accept the procedure when it pops up. Strategy *S2* is a bit more extensive. Before accepting a procedure, the proposed procedure is checked on its safety. If the solution is safe, the trajectory will be accepted. In strategies *S3* and *S4*, the proposed procedure will not only be checked for its safety, but will also be checked for performance indicators such as flightpath length or robustness. The difference between *S3* and *S4* is in whether one or more performance indicators will be checked.

Strategies *S5* and *S6* resemble a deeper form of analysis. In *S5*, other feasible options will be compared to the proposed solution. This strategy starts with asking what other trajectories would be of interest. Then the safety and performance needs to be found for these trajectories. Finally, the relative performance could be compared and decision on the acceptation of the solution can be made. In *S6*, not only the proposed procedure itself is questioned, but also how the cost values are created will be under investigation. Additional info of the physics that contribute to the build up of the solution, will be shown. Using that information, it should be considered whether the cost values are logical.

More strategies could exist. These six strategies are just a subset of strategies that were found to be of interest for this design. Designing for the “one right way” should be avoided. The goal is to facilitate a large variety of strategies [77].

A.5.4. Social Organisation and Coorporation Analysis

During this phase, the allocation of tasks between human and machine is set. Generally, this phase is integrated into the decision ladder of the control task analysis. That was also done in this CWA. In Figure A.21, all relevant steps of the ladder have a color. Blocks with no color are skipped. The yellow colored boxes are performed by humans and the grey boxes by automation. There are also knowledge states that are shared between automation and human. These are colored green.

The ladder starts with a grey box. Automation will activate the user by notifying her of a proposed procedure. The results of this is a shared state of knowledge that the human will need to perform the task. Observation and option generation are both done by the human and colored yellow. The options that need to be evaluated, are shared knowledge as the human has generated them, but the computer will need knowledge of these options in order to compute their performance. The knowledge about the performance of the options is again shared and subsequently used to reason whether to accept the procedure or not. Finally, the human will execute confirmation or cancellation of the proposed solution.

A.5.5. Worker Competence Analysis

The final phase of the CWA is the worker competence analysis. This phase is about the psychological constraint that are applicable to design [39]. The central question is: “How can human actors be

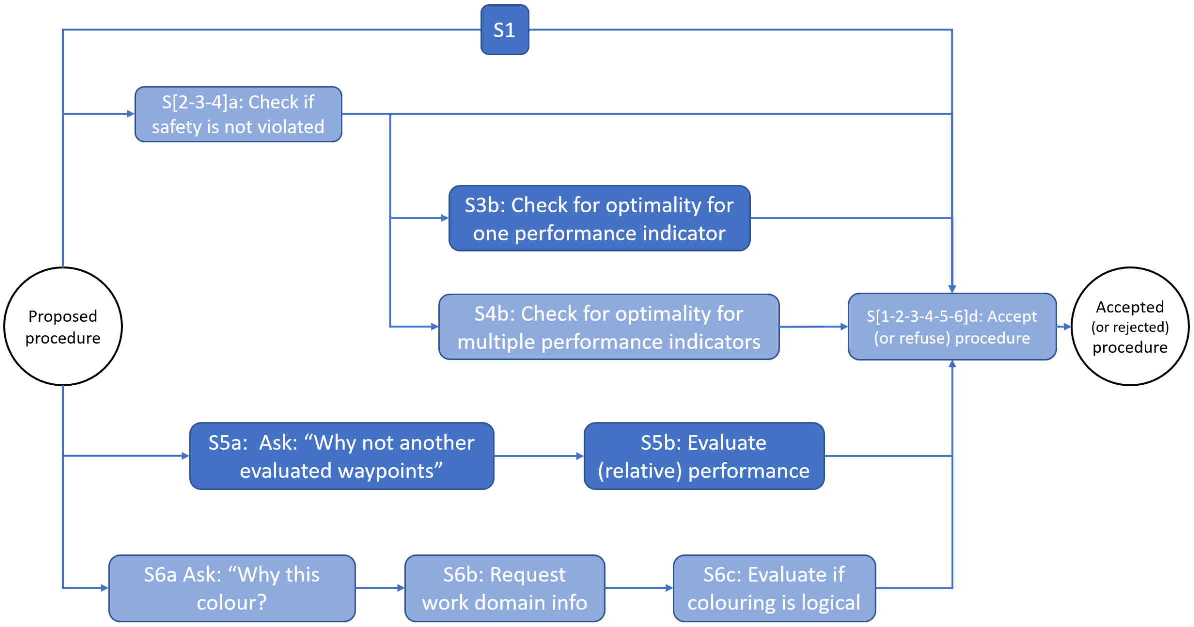


Figure A.22: Strategy analysis flow diagram of the supervisory task.

supported in their task?" [39]. The SRK Taxonomy of Rasmusssen [61] as decribed in Section A.2.4.1 is used.

The control task steps that are described in the Control Task Analysis will each be supported by the interface. In Table A.3, three columns are dedicated for SBB, RBB and KBB for each step of the decision ladder. These columns contain the information that the interface could/should show to support the human actor during their work. Interface should support mainly SBB and RBB for fast response time. However, a seamless transition to KBB should be possible for problem solving.

A.6. Interface Design

This chapter will present the designs for four levels of transparency as well as how these designs are created. The first section will give an overview of the design process. It describes how elements for explanations were identified and it is stressed that interface design is an iterative process. The second section will show the design for the four levels of transparency. A description of each design has been provided as well.

A.6.1. Process

In this section, the overall design process is described. Note that the chapters considering transparency, algorithm workings and the cognitive work analysis have formed the basis for this design process.

A.6.1.1 Elements of explanation

The first step in creating the design was identifying elements that could explain parts of the algorithm. The AH, section A.5.1, was taken as a basis to discover these elements. Table A.4 gives an overview of all those elements. They have been ordered according to increasing transparency.

The start level is showing the proposed trajectory and the constraints that drive the work domain. This will show the underlying rationale that affects automation as well. The next step is the visualisation of the evaluated and possible states by visualising the grid, state change and interval strategy. One level higher the reachable states are visualised. Because the blocked stated are the inverse of the reachable states, these elements were considered to be in the same level. The fourth level of transparency includes the visualisation of the cost values either visually or interactively. For all transparency levels, interface designs were created. The next section will give an overview of these designs.

Table A.3: The steps of the decision ladder in Figure A.21 are explained in more detail in this table. The last three columns state what information will support SBB, RBB and KBB.

Number	**Information processing step**	**Resultant state of knowledge**	**SBB**	**RBB**	**KBB**
1a, 1b	User is activated	Knowledge that supervision is required	Trajectory pops up, chime	color or something that tells user that action is required!	Text that tells that a trajectory is proposed. For example: “New resolution proposal”
2a, 2b	Observe proposed trajectory by automation	Knowledge of the procedure and related performance proposed by automation	Perceive procedure as a series of waypoint with connecting lines	Red colors indicate unsafe operation, green colors indicate safe operation. The darkness of green gives clues to the optimality of the solution	
3a, 3b	Come up with other options for trajectories	Knowledge of what other options would be logical	Show (evaluated) grid points and the limits of the evaluated solution space as imposed by state change interval strategy.	Color coding could be used for finding “interesting areas”. Additionally, operator heuristics for how the operator would solve conflict. Supported information are reachable states and time distance to collision.	Reason potential trajectory based upon geospatial knowledge of destination and obstacles and the state change capabilities of the aircraft
4a, 4b	Check options with system goals: Safety, Efficiency, Productivity	Knowledge of performance of other options	Perceive performance from color shading of points or areas	Use heuristics to interpret overall performance using performance indicators	Reason from work domain constraint the underlying rationale for a particular cost. For example: Showing TSR-like interface with unfeasible and feasible regions together with optimality within feasible regions
5a, 5b	Check if other options are better	Knowledge of whether to confirm proposed procedure or not	Color coding performance indicator; This will indicating whether performance is relative good or not. Furthermore, color coding of cell gives rough idea of relative performance.	If other options cost is higher accept procedure.	Reason about what aspects are (relatively) important: Distance, Delay, Adaptability/ Robustness. For example by showing the contributions of each aspect to the total cost.
6a	Execute solution if confirmed		Press button to execute and get audio or visual confirmation		Reason if aircraft reacts in a logical manner according to proposed procedure.

Figure A.23: Brainstorm for generating explanation visualisation ideas.

Note that for each element, it has been indicated what link in the AH it tries to visualize. Furthermore, the triggered question that it wants to explain is also presented in the table.

A.6.1.2 Iteration

In order to come up with ideas of how the interface should look like, a brainstorm was held with 3 TU Delft students from different faculties. The main topic of the brainstorm was finding representations of the crucial elements of the algorithm. Figure A.23 shows the collection of ideas and sketches. A selection of these ideas was further developed and constructive feedback was provided by staff of the "Control and Simulation" department of the faculty of aerospace engineering of the TU Delft. This has led to the design presented in the next section.

Designing the interface is a highly iterative process. The current design has already been iterated several times. It is expected to be subject to change even more during the phase in which the interface is programmed. The same holds for the ordering of transparency levels. When the interface is implemented, it can be further investigated whether these levels are in correct order. Finally, more extensive evaluation and optimisation can be done using the display design principles as stated by Wickens et al. [83].

A.6.2. Interface Designs

In this section, the designs for the 4 levels of transparency are presented. The designs where made for a situation where a faster aircraft is in conflict with the aircraft in front of it. They are based upon Tables A.3 and A.4. The algorithm is solely optimizing for shortest path. The design might change if another cost strategy will be used. The concepts of "What if" and "Why not" explanations are integrated into the design as is suggested in chapter A.4. The use of progressive disclosure is limited. The division of the explanation in transparency levels limits the possibilities for progressive disclosure. Interaction has been used in levels 3, 4 in the current designs.

The baseline is showing the constraints of the work domain. The general idea is to show more constraints that are considered by the algorithm, for higher levels of transparency. The solution space under consideration will thus be narrowed down and finally the algorithm's interpretation of the final solution space is presented in the last level. For each level of transparency, the operator will dive deeper into the algorithm's rationale. A detailed explanation of all levels will be further discussed now.

Figure A.24 is a design for the first level of transparency. In this level the outcome of the algorithm is visualised: a trajectory proposal. The white line forms the trajectory and the white hexagonal point

Table A.4: Elements that contribute to explanation. It has been indicated in which level of transparency they are grouped, what they are explaining and how they do that. Also the links of the AH, see Figure A.20, that correspond with the element has been indicated.

ID	Type	Element	Link in AH	Level	Explanation for:	How:
0	-	No additional support	-	-	Nothing	-
1	Outcome	Clearances	Locomotion - routing	1	What is the immediate action?	Showing the heading and speed change for new trajectory proposal.
2	Outcome	Track	Locomotion - routing - waypoint	1	What is the proposed trajectory?	Showing track and intermediate waypoint.
3	Input/Process	Work Domain Constraints	Economy - Routing - Cost function	1	What constraints are part of this work domain? What is the underlying rationale of automation?	Show the work domain constraints. Example is the TSR with equal distance ellipses for shortest path.
3	Process	Grid	Routing - Grid	2	How does the algorithm discretizes the work domain?	Showing grid that automation uses.
4	Process	Evaluated states	-	2	What points are evaluated?	Group the cells that are evaluated.
5a	Process	State Change	State change strategy - Routing	2	What are the allowed state changes of the algorithms?	Showing the space that is available within the state change constraints.
5b	Process	Interval	Interval strategy - Routing	2	At what time intervals will the algorithm look?	Group the dots that correspond to the same time interval.
6	Process	Reachable states	-	3	What are the states the aircraft could travel to?	Color the cells that are reachable.
7	Process	Obstacles	Obstacle - Routing	3	What evaluated aircraft states are blocked due to an obstacle?	Color the cells that are blocked.
8	Input	Obstacle causation	Obstacle - Obstruction	3	Why are these blocked states blocked?	Highlight reason for blockage (e.g. aircraft).
9	Process	Cost values	Cost function - routing	4	Why not another trajectory? What are interesting spots for waypoint placing?	Use color scale to indicate relative performance.
10	Process	Interactively exploring cost values	Cost function - routing	4	Why not another trajectory?	Use color coded value to resemble relative performance of both trajectories.

resembles the chosen intermediate waypoint. Additionally, the TSR, see section A.2.4.3, will be presented during this level. This will show the constraints from the work domain that drive automation. The green ellipses indicate that flying further away will be less efficient. That is the core principle for the cost function of shortest path.

In level 2, see Figure A.25, more information about the inner workings of the algorithm is given. The information retrieved from the TSR will still be presented. The grid strategy is shown by a grey grid. This will help interpreting how the algorithm discretizes the work domain. Furthermore, the blue fading lines indicate the maximum and minimum angle the aircraft is allowed to deviate from its current path. The bands with black lines resemble the possible states the aircraft could have at a time interval. This is similar to the visualisation of the state change strategy in Figure A.16. The bands are limited by the maximum allowed angle deviation,the boundaries of the grid and minimum and maximum change in speed. The cells that overlap with the state change strategy band will remain fully transparent. These are the cells that will be further evaluated by the algorithm. All other cells will be given a dark overlay and will not be evaluated at all. It was chosen to remove the grid lines in the unevaluated areas in order to prevent display clutter. Therefore, the TSR can be seen more easily and the focus is automatically placed on the cells that are evaluated by the algorithm. This design adds information about the grid strategy, state change strategy and interval strategy of the algorithm.

Figure A.26, shows the design for level 3. The level builds upon the previous layer by showing which intermediate waypoints are possible. Cells with transparent red overlay are blocked by either a static or dynamic obstacle. Hovering over the blocked cells with the cursor will show the causation of the blockage by highlighting it. Cells with light blue overlay are possible to be reached by the aircraft. However, it would not be possible to get to the end point within the given time constraints. Finally, the full transparent cells indicate the feasible trajectories. These waypoints will be reachable from the begin state and they will also allow for reaching the endpoint within the given constraints. Hovering over the cells will show the operator in the top right corner if the cell is either unevaluated, unfeasible, blocked or feasible.

Level 4, Figure A.27, will not only show which trajectories are possible, but will also show which trajectories were found to be more and less favourable by the algorithm. This is done by showing the cost values that the algorithm uses in order to make a decision. These are represented by a green color scale. In this version, darker green is less favourable. Another way for the operator to find out what the cost values of the cells are, is by clicking on them. If they click on a green cell, the top right corner will reveal a percentage drop or increase with respect to all aspects of the cost function of the proposed trajectory. The dark blue line resembles the alternative trajectory chosen by the operator. It is chosen to make the evaluated cells fully opaque in this level.

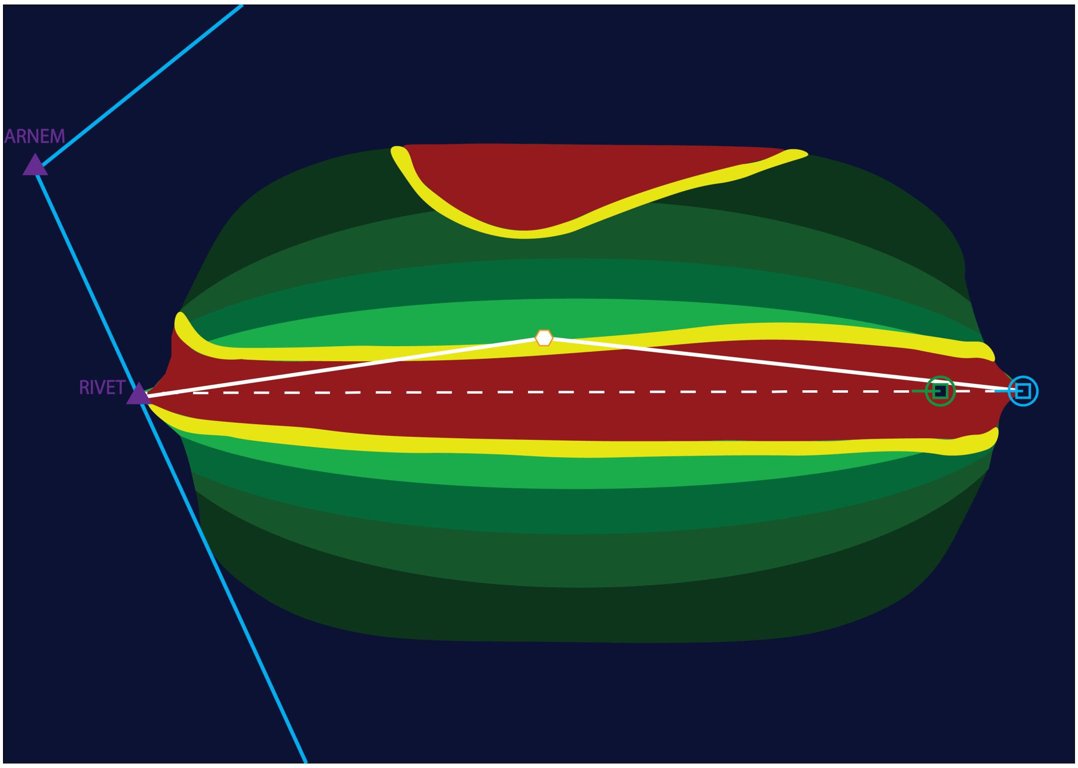


Figure A.24: Transparency level 1: Showing the TSR with the proposed automated trajectory. Waypoints placed in red areas will result in unsafe operation. Green areas will result in safe operation. Yellow areas indicate the margin. Additionally, green ellipses have been introduced in order to show the constraints that drive the cost function.

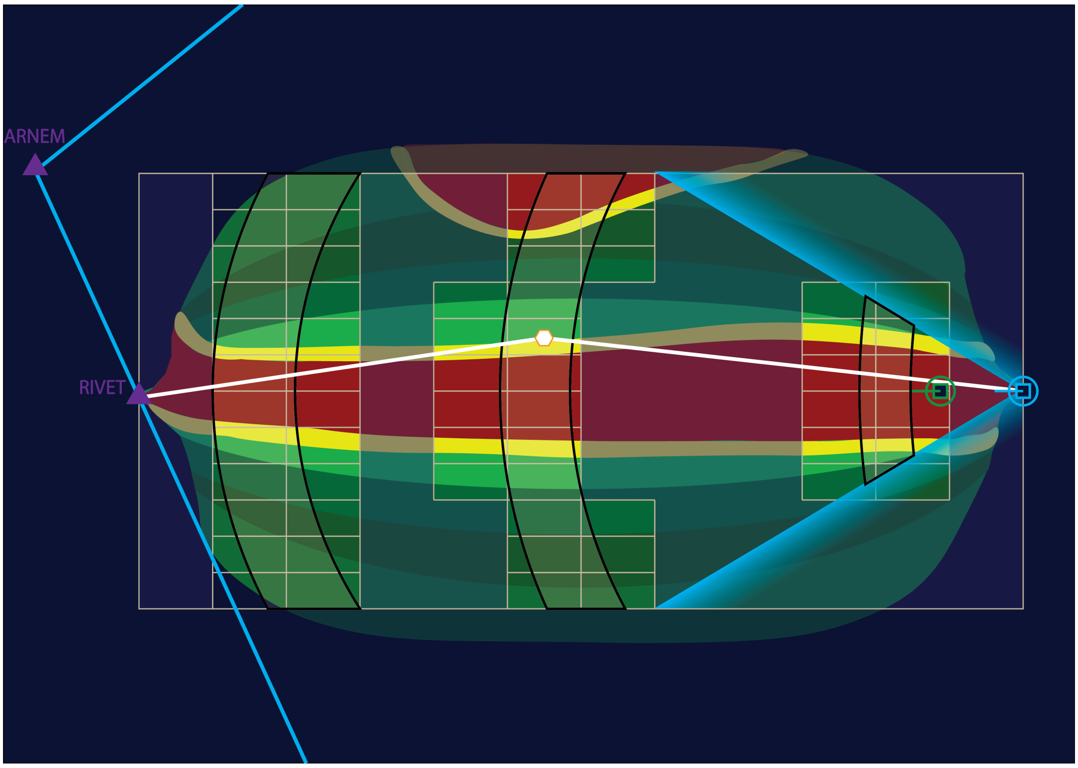


Figure A.25: Transparency level 2: Showing the cells that are evaluated. Cells that are not evaluated have a dark overlay and no grid lines. The state change strategy is indicated by bands with black edges. Each state-change band is a interval. Grid strategy is shown by the grey lines. The maximum allowable heading deviation are indicated by fading light blue lines.

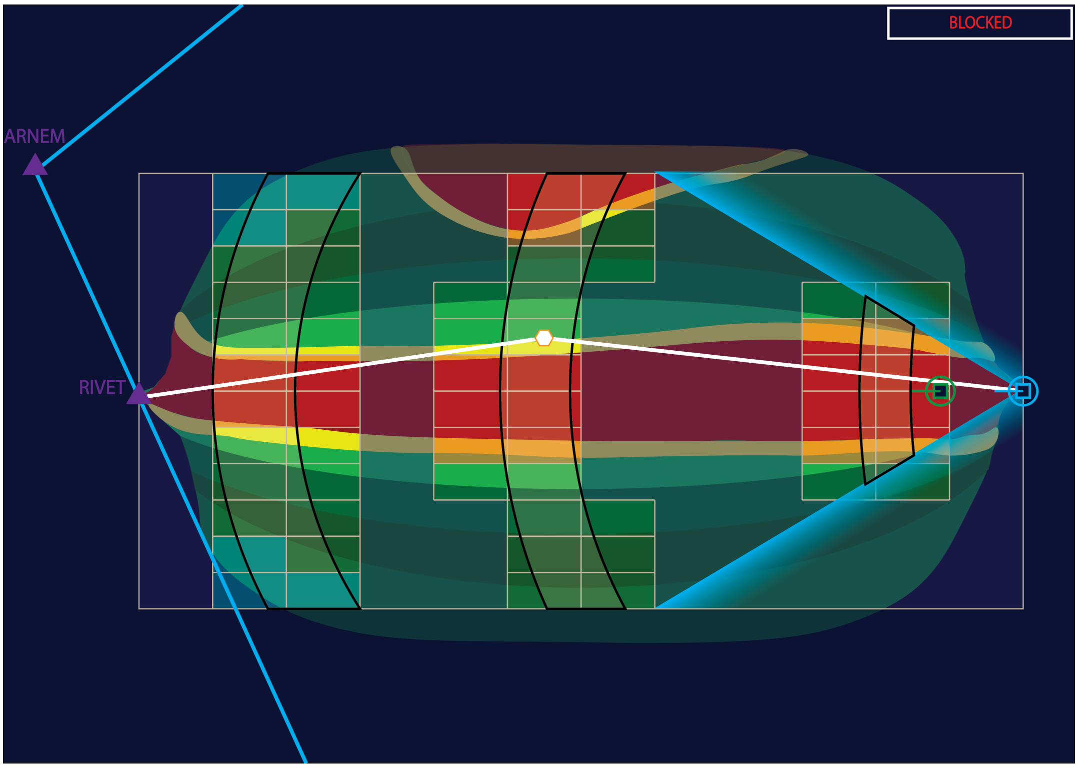


Figure A.26: Transparency level 3: In this level more information about the evaluated trajectories. There are 4 options: unevaluated(dark), infeasible (light blue), blocked (red) or feasible(full transparent). Hovering over the cells will show the operator in the top right corner what state belongs to the cell.

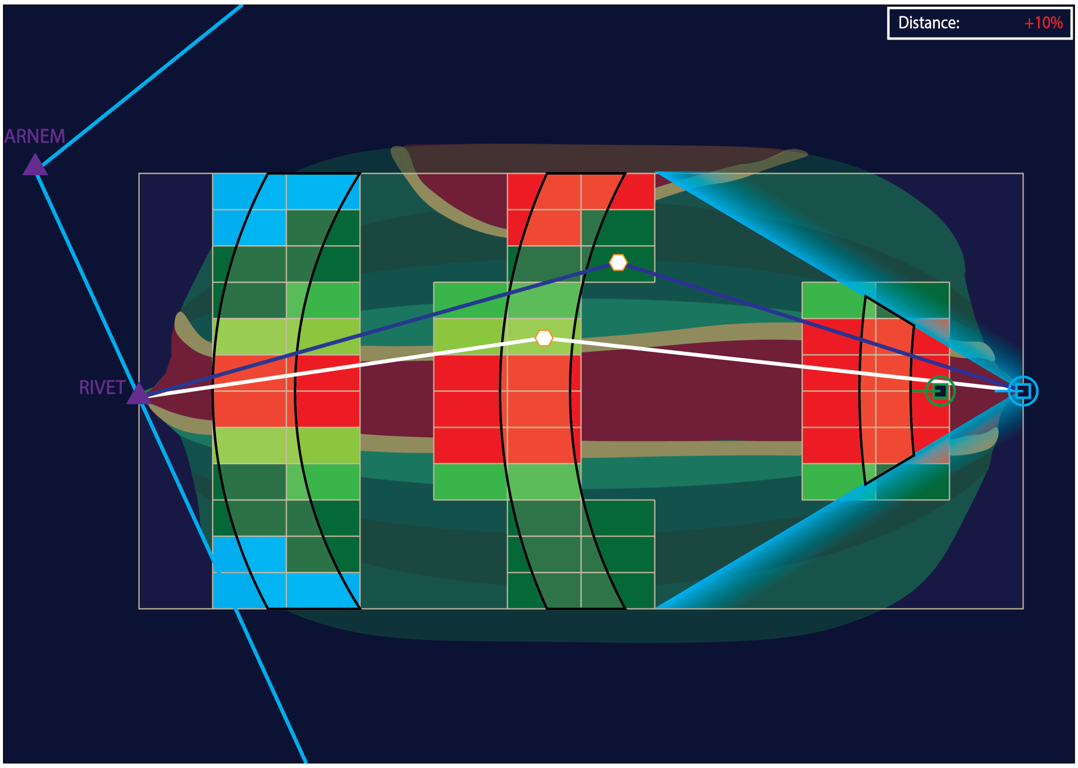


Figure A.27: Transparency level 4: The feasible cells are colored according to the cost function. Light green cells are more favourable than dark green cells. All evaluated cells are made fully opaque. The operator can click on a cell in order to evaluate its performance. The performance will be shown in the top right corner.

B

Scenario Design

This chapter gives an overview of how the scenarios were designed. First, the traffic situations are discussed in Section B.1. An important part of the scenarios are the questions. These are discussed in B.2. Finally, the designed scenarios are shown in Section B.3. Figures of all experiment scenarios is provided in that section as well.

B.1. Traffic Situation

The goal was to create scenarios that contain a large variety of aspects of the algorithm. At the same, time it was desired to have scenarios of equal difficulty. Furthermore, an important objective was to create scenarios that were not obvious from the start such that the curiosity of participants was triggered.

Variety of aspects of the algorithm The experiment should test a large variety of aspects of the algorithm in order to test the overall understanding of the algorithm. The starting point for this objective is the AH. The AH shows the meaningful relationships between elements in the work domain and therefore gives an important indication what the operator needs to be able to understand. The scenarios were thus designed such that they show a large variety of elements and links of the agent part of the AH. This also facilitates better use of the interface as more interface elements are required for creating understanding throughout the experiment.

Five core types of scenarios were chosen. These are related to the grid strategy , interval strategy, obstacle strategy, state-change strategy and cost strategy. The settings changed in these scenarios were cell size, time interval, separation criteria, time constraint and the strategy. These changes in settings were then the dominant reason for the algorithm to choose a certain trajectory that was not necessarily the most logical option to choose. Next to these five core types, also other elements of the AH were asked for in Question 4. This is more thoroughly explain in Section B.2.

Equal Difficulty Creating scenarios of equal difficulty was achieved by a number of choices. First of all, the sector shape of all scenarios was the same. The sector used was the MUNSTER sector of Maastricht Upper Area Control (MUAC). This sector had an interesting shape, which made it possible to make a large variety of traffic situations containing both long and short routes. Furthermore, it was chosen to use 2 to 3 aircraft. By not having a large variety in the amount of aircraft, equal scenario difficulty was stimulated as well.

Some scenarios that contained two aircraft, also had a restricted airspace. Choosing for a scenario that had a restricted airspace, also makes sense when looking at the AH. The agent makes a distinction between *static* and *dynamic* obstacles. Static obstacles are objects like restricted airspaces and dynamic obstacles are objects such as other aircraft in the airspace. Having a scenario with an restricted airspaces thus represents a better representation of the AH in the scenarios. An overview of the structure of the scenarios can be found in Table B.1.

Table B.1: Overview of scenarios used in the experiment.

ID	Setting changed	Strategy	Aircraft	Restricted airspace
CEL	Cell size	SP	2	Yes
INT	Time interval	LSM	3	No
SEP	Separation criteria	SP	2	No
TIM	Time constraint	LSM	2	No
STR	Strategy	SP	2	Yes

Table B.2: Type of questions and how they relate to SAT.

Question No.	SA-Agent Based Level	Level of understanding
Question 1	Level 1	Plans & Goals
Question 2	Level 2	Algorithm reasoning
Question 3	Level 2	Algorithm reasoning
Question 4	Level 3	Projection

Non-trivial scenarios The last objective was to create scenarios that were non-trivial. Given the condition stated in the previous two paragraphs, it was mainly trial and error to find a correct composition of objects such that the scenarios were not trivial. In order to make strategies non-trivial, the two strategies, largest separation margin and shortest path, were alternated.

B.2. Questions

An important part of the scenario are the questions. Per scenario four questions are asked and they are related to the three level of SAT [14]. By asking these four questions, different levels of understanding were tested. These questions are followed up by a confidence question that indicates how confident the participants are about their answer. The confidence questions have a scale from 0-100. Table B.2 shows how the four questions are related to the SAT framework.

The following part of this section will go into more detail about the four questions. One important thing to know that applies to all of these questions is that of the answer option “Cannot answer”. All question have this answer option. In case participants think they have insufficient information to answer this question, they should use “Cannot answer”. Participant are told to interpret the confidence level in this situation as how confident they were that they could not answer the question with the information available to them.

B.2.1. Question 1

Question 1 (Q1) was related to the first level of SAT. At this level, the plans and goals of the agent should be clear. Therefore, this questions asked about the goal and plans of the agent. Two types of these questions were created and both questions with their answer options can be seen Tables B.3 and B.4. These questions were found very easy and were always answered correctly with high confidence. The participant was allowed to choose one option only.

Table B.3: Q1a: This questions is related to the goal of automation.

Question:	What waypoint is automation trying to reach?
Answer 1	Waypoint 1
Answer 2	Waypoint 2
Answer 3	Waypoint 3
Answer 4	Waypoint 4
Answer 5	Waypoint 5
Answer 6	Cannot answer

Table B.4: Q1b: This questions is related to the plan of automation.

Question:	How does automation propose to fly?
Answer 1	Fly direct
Answer 2	Fly to the left
Answer 3	Fly to the right?
Answer 4	Cannot answer

Table B.5: Q2: This questions asks the participant about the strategy of automation and thereby tests the understanding of the automation's reasoning process.

Question:	What strategy is automation using?
Answer 1	Shortest path
Answer 2	Largest separation margin
Answer 3	Cannot answer

B.2.2. Question 2

The second question was connected to SAT Level 2. In SAT Level 2, the algorithm's reasoning process should be made clear. The strategy of automation was found to be one of the core elements driving the reasoning process of automation. Therefore, a question was asked about the strategy in every scenario. Table B.5 shows the question and possible answer options. The participant was allowed to choose one option only.

B.2.3. Question 3

The third question also relates to SAT Level 2 and asks the participant about the reasoning process of automation. This question does not directly asks about the strategy, but rather focusses on the dominant reason why another point has not been chosen. The correct answers of this question are always directly related to the type of scenario.

Table B.6 shows the question with all answer options. Not all answer options will be given for each scenario. A total of 6 answer options are presented to the participant in each scenario. The "cannot answer" option will always be present. The scenarios were made such that one answer is clearly the dominant reason for not choosing another point. The participant was allowed to choose one option only.

B.2.4. Question 4

The fourth questions deals with SAT Level 3. The goal is to measure how well the participants is able to make predictions about the algorithm. These questions require participants to think further then simple if-then rules and requires primarily KBB for answering.

This question asked how another point could be made feasible. A similar set of answer options

Table B.6: Q3: This questions asks the participant about the most important constraint that drive the automation's reasoning process.

Question:	Why did automation not propose to fly pass point XXX?
Answer 1	Strict separation criteria
Answer 2	Strict time constraint
Answer 3	Time interval
Answer 4	Maximum heading lines
Answer 5	Heading increments
Answer 6	Optimizing for shortest path
Answer 7	Optimizing for largest separation margin
Answer 8	Tile sizes
Answer 9	Cannot Answer

Table B.7: Q4: This questions asks the participant to make prediction about the algorithm by asking what settings should change in order to make other points considered feasible. Note that multiple answers could be selected.

Question:	What settings of automation should change such that point XXX will be feasible?
Answer 1	Tile sizes
Answer 2	Time constraint
Answer 3	Time interval
Answer 4	Maximum heading lines
Answer 5	Heading increments
Answer 6	Grid size
Answer 7	Cannot answer

were presented to the participant as in Table B.6 with the exception of options related to strategies (note that only feasibility is asked) and separation criteria (this was found to ambiguous). The participants needed to select all answer options that were required to make a particular point considered feasible. Multiple answer could thus be given. The question was only counted correct if all correct options were chosen. The participants were asked to interpret confidence as their certainty that their given answer is complete. Table B.7 shows the question and answer options.

Participants found this question the most difficult one. Some of them found it even ambiguous. Results, see Appendix E, correspond with these statements.

B.3. Scenarios

This sections gives an overview of the scenarios used. First the scenarios in the training phase are discussed. Afterwards, the five scenarios of the experiment are shown.

B.3.1. Training

During the training phase, the participants get acquainted to the interface, task and capabilities of the agent. All types of scenarios are present in the training phase. In total nine scenarios were used to prepare the participants. The scenarios are explained in groups.

Interface explanation The first two scenarios were used to explain the interface. In these scenarios, there is a simple traffic situation of two aircraft and a restricted airspace. First, all elements are explained using the bottom-up approach in Scenario 1. This is done by means of a transparency slider in which the investigator and participant could easily go up and down in the level of transparency. In Scenario 2, the same explanation is performed for the top down approach. It was made clear that the TSR and agent could have a different view on the work domain and thus visualisation might not always be in correspondence with each other. For example, the agent might be a bit stricter in judging if something is a conflict or not compared to the TSR.

Questions Scenario 3 is dedicated to explaining questions and the overall procedure. There is no time limit in this scenario. Investigator and participant slowly proceed through all questions and answer options. Participants have the opportunity to ask questions if anything is unclear. Furthermore, the procedure within a scenario is discussed. This entails that after answering questions a confidence question pop-ups that ask for their certainty regarding the answer they have given. If all questions and confidence ratings have been completed, the next transparency level was shown. This continued till Level 3. After answering that question, the scenario was completed and the next scenario was loaded.

Bottom-up The participants continues the training phase with Scenarios 3, 4 and 5. These scenarios use the bottom-up approach. The scenarios had the separation criteria, strategy and tile size as changed settings.

Top-down After three scenarios in bottom-up approach, the final three scenarios of the training phase were done in the top-down approach. The settings changed in these scenarios were separation criteria, time interval and time constraint.

B.3.2. Experiment

During the measurement phase, the participant did two rounds of five scenarios each. The two rounds differed in the transparency direction. This was either top-down or bottom-up. The five scenarios were rotated 90 degrees to prevent recognition of the scenarios in the second round. All five scenarios can be seen in Figures B.1 through B.5.

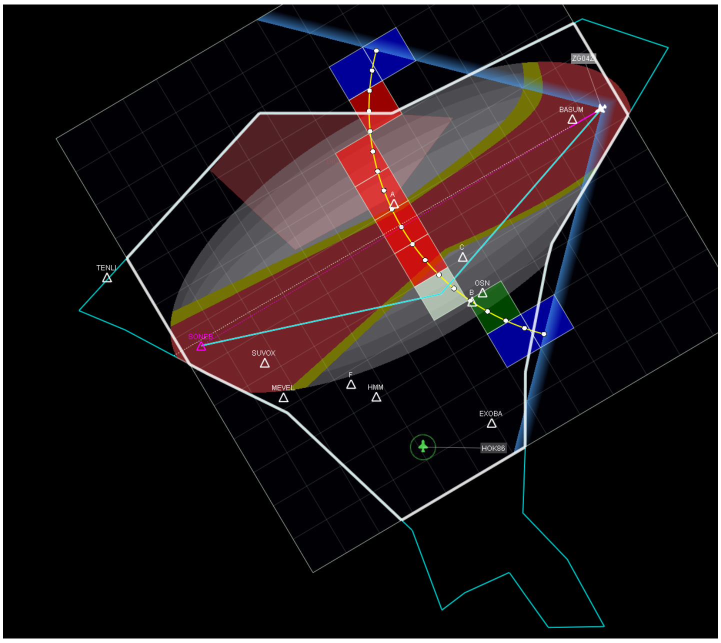


Figure B.1: Scenario CEL (Cell size) at full transparency.

Scenario CEL (Cell size) In Figure B.1, scenario CEL can be seen. The first question in this scenario was of type Q1b. The answer to that question is “Left” as the automation proposed to steer the aircraft to the left. The answer to Question 2 is “Shortest path”. Note that there is another cell that is feasible, but that cell resemble a trajectory with an even longer flight path.

Question 3 asked what the dominant reason was for not choosing point A for re-routing. The answer to this question is “Tile size”. Due to the large tile sizes both cells to the sides of point A are considered infeasible, because their center points fall within an unsafe field of travel. Furthermore, note that right next to Point A an state-change dot is drawn. If the tile size shrinks, the center point of the tile of point A will fall within the safe field of travel and because there is also a state-change dot at that location, that cell will be under further evaluation and will be considered feasible. Therefore, “Tile size” is the dominant reason why point A is not chosen.

Question 4 asks what settings of automation should change in order to make point C considered feasible. The correct answer to this question is “Tile size” and “Time interval”. Point C lies in the middle of two tiles. Therefore, if any of those cells would be chosen, the trajectory will still not fly directly to Point C. Furthermore, the time interval needs to be changed, because the algorithm is not looking for a solution in the area of Point C. Changing the time interval would change the location of the state-change band and thereby the area of Point C can be evaluated.

Scenario INT (Time interval) In Figure B.2, scenario INT can be seen. The first questions asked what the target waypoint of automation is. The correct answer for this questions is “EXOBA”. The correct answer of Q2 is “largest separation margin”. Note that there are feasible cells with much shorter routes in this scenario.

The correct answer to Q3 is “time interval”. It is asked why point B is not chosen. Note that there is no state-change band close to point B. Therefore, the time interval should change in order to make the area around point B under evaluation.

Q4 asked why point C is not feasible. The correct options for this question are “Maximum heading lines” and “time interval”. Note that point C falls behind the maximum heading lines. In order to make

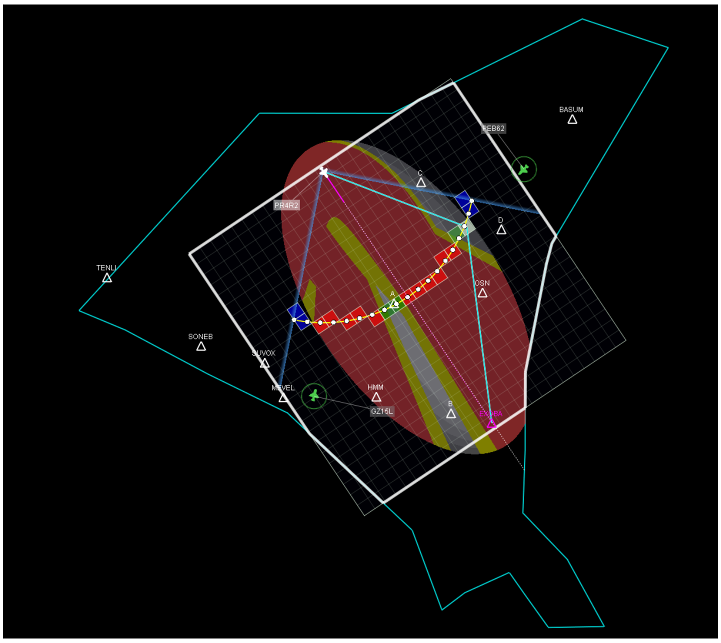


Figure B.2: Scenario INT (Time interval) at full transparency.

Point C evaluated, the maximum heading allowed to deviate at the start should be increased. Furthermore, no state-change band is close to Point C. Therefore, the time interval needs to be changed as well.

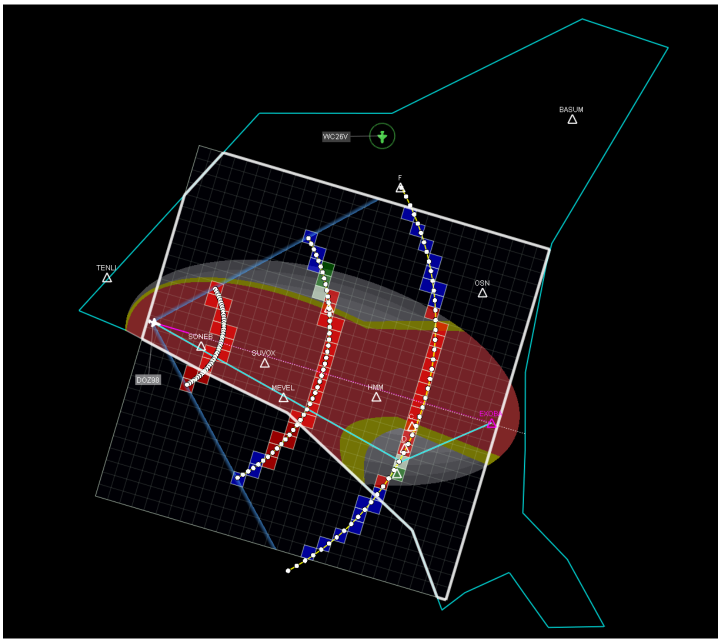


Figure B.3: Scenario SEP (Separation criteria) at full transparency.

Scenario SEP (Separation criteria) In Figure B.3, scenario SEP can be seen. The first question in this scenario was of type Q1b. The answer to that question is “Right” as the automation proposed to steer the aircraft to the right. The correct answer for Q2 is “Shortest path”. This can be seen by looking at the different shades of green. Cells closer to the direct path are coloured lighter green than cells that are further away. Therefore, the agent aims to optimize for shortest path.

Q3 asks why Point D was not chosen. The correct answer to Q3 is “Strict separation criteria”. Note that a lot of cells in the safe field of travel are colored red. This indicates that the algorithm is more strict in determining if a cell would lead to a loss of separation or not.

Q4 asked what settings should change in order to consider Point F as feasible. The correct answers to Q4 are “Grid size” and “Time constraint”. Point F is outside of the grid. Therefore, the size of the grid should increase in order to consider point F. Furthermore, the time constraint needs to be loosened. Otherwise the trajectory to point F would require too much time to complete and will therefore not be considered feasible.

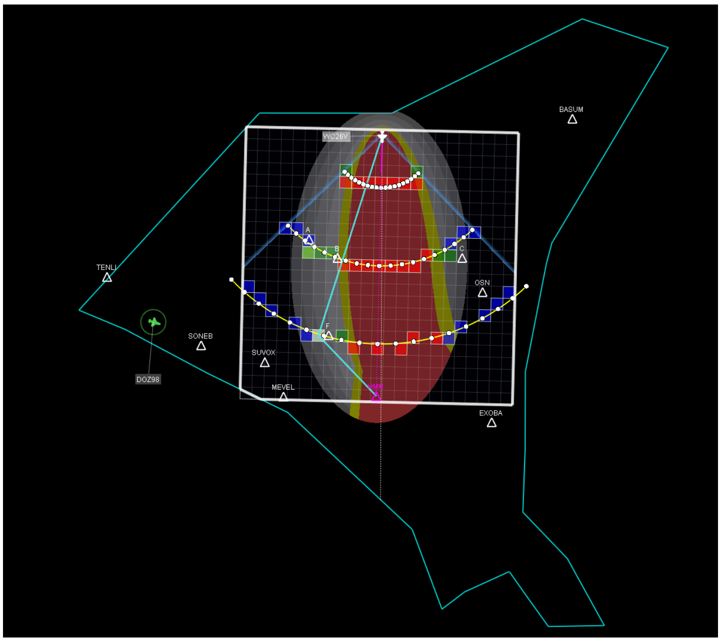


Figure B.4: Scenario TIM (Time constraint) at full transparency.

Scenario TIM (Time constraint) In Figure B.4, scenario TIM can be seen. The first question in this scenario was of type Q1a. The answer to that question is “HMM”. The correct answer for Q3 is “Largest separation margin”. Note that feasible cells get lighter color (and thus lower costs) if they are further placed from the unsafe field of travel. Therefore, the algorithm optimizes for largest separation margin.

Q3 asked why Point A was not chosen. Note that Point A lies within a blue cell. The meaning of a blue cell is that it is not considered feasible, because the delay is too large. Therefore, the correct answer to Q3 is “Time constraint”.

Q4 asked what settings of automation should change such that point F would be considered feasible. Note that Point F falls within two feasible cells and is on the state-change band. The cell is not under evaluation, because it contains no state-change dot. Therefore, the correct answer to Q4 is “Heading increments”. If the heading increment becomes smaller, more state-change dots will be present on the state-change band and thus will Point F be evaluated.

Scenario STR (Cost strategy) In Figure B.5, scenario STR can be seen. The first question was of type Q1a. The correct answer to that question is “SONEB”. The correct answer of Q2 is “Shortest path”, because cells that are further away from the direct path are considered to be less optimal and are coloured in darker shades of green.

Q3 asked why Point B was not chosen. Point B is considered feasible already. The dominant reason for not choosing Point B is thus that the algorithm is optimizing for largest separation margin.

Q4 asked what settings of automation should change such that Point HMM would be considered feasible. The correct answers are “Heading increments” and “Time constraint”. Note that Point HMM is surrounded by blue cells. Thus if that cell would be under evaluation, the time constraint should definitely change. Furthermore, the heading increments should increase such that the cell of Point HMM contains a state-change dots and will thus be further evaluated by the algorithm.

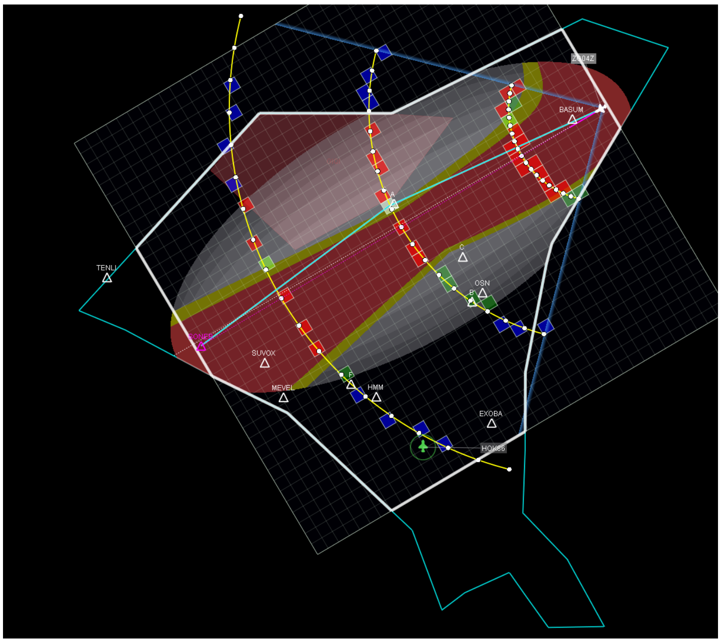


Figure B.5: Scenario STR (Cost strategy) at full transparency.

C

Experiment Briefing

Effect of Agent Transparency in Air Traffic Control

BRIEFING DOCUMENT, DATE: 20-04-2021

1. Introduction:

There is a general trend for increasing the level of automation in the ATC System. The dominant reason for this is to make the ATC system more efficient and safe. At the same time, humans are expected to play a central role in such as automated ATC system. Therefore, humans and machines need to be able to collaborate. An important aspect of this collaboration, will be maintaining sufficient understanding of what automation is doing and why. This research aims to investigate how visualizations can support operators in understanding automation in ATC.

This research focuses on a rerouting task as part of managing airspace disruptions (e.g., traffic conflicts, no-fly zones and weather cells). Figure 1 gives an overview of this situation. The blue aircraft is planned to fly across the sector to the exit waypoint (the triangle on the right). However, the direct route is obstructed due to a perturbation, which is the red circle in the middle. This perturbation could be induced by bad weather or by other aircraft traversing the sector. The task of automation is to find a safe route to the exit waypoint. In the example in Figure 1, automation has chosen the orange route. This may leave the human air traffic controller with the question why it did not choose another route such as the blue one. In the experiment, you will act as an air traffic controller and your task is to find out why the automation has chosen a particular route.

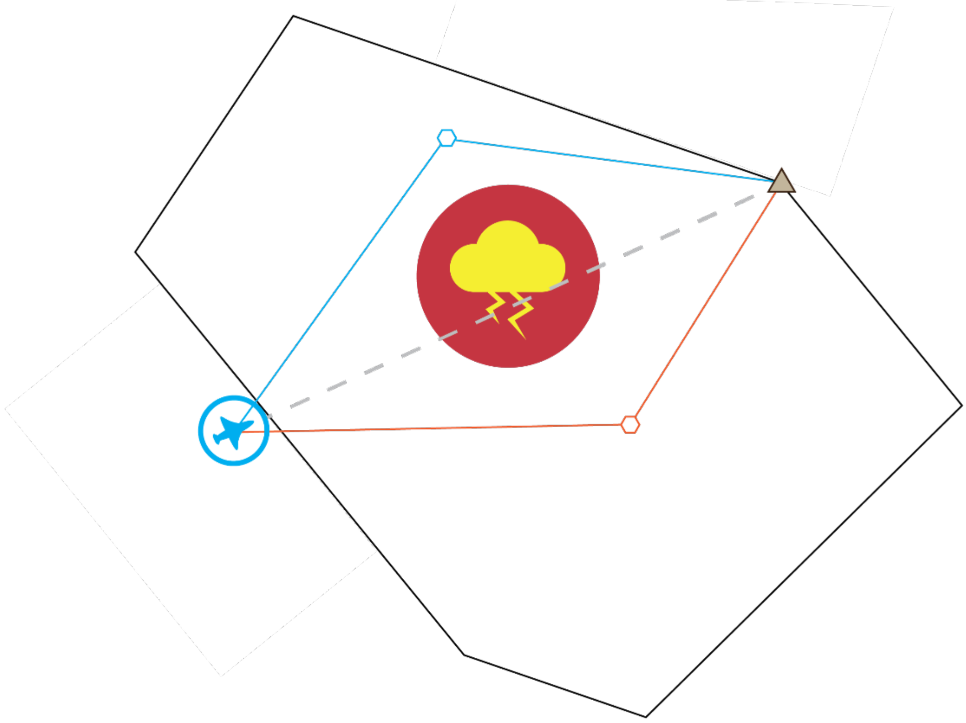


Figure 1: Rerouting situation. Bad weather makes it necessary to reroute. Automation has proposed the red route. But why did it not choose the blue route? Understanding of automation's inner process is required to answer that question.

An experiment is performed in order to investigate how the automated agent can be made more transparent. The experiment is done in an ATC simulation. Static traffic scenarios in which automation has proposed a new route will be shown. You need to answer questions about the choice of automation within a certain time limit. Detailed information about the experiment can be found in Section 3.

2. Interface

The Travel Space Representation (TSR) was chosen as baseline in order to provide insights into the traffic situation. Additional interface elements are used to show information about the automation's decision process. In the first part of this section, the TSR is explained. The second part gives a general overview of the additional interface elements.

TSR

The TSR supports rerouting of aircraft using intermediate waypoints. The TSR shows the space in which a feasible solution can be found within the constraints of the problem. In this research, it is assumed that aircraft fly with constant speed.

Figure 2 shows an example of the TSR interface. The basic form of the TSR is an ellipse. The mathematical property of the ellipse indicates that all points on it will result in exactly the same track length. Therefore, considering that speed is constant, each ellipse resembles a certain amount of delay. In Figure 2, route A and B result in equal delay. The boundary of the TSR is determined by the maximum allowable delay. An indication of aircraft delay is shown in the TSR by the different shades of grey. Darker grey ellipses indicate a larger delay and travel distance than lighter ones.

Inside these ellipses, three types of areas can be found. Grey areas resemble the safe field of travel. In these areas, the placement of an intermediate waypoint will result in safe travel. The red areas indicate the restricted field of travel. Placing an intermediate waypoint in these areas will result in a trajectory that will lead to a loss of separation. This means that two aircraft will have less than 5 nm separation. Finally, the yellow areas show a safety margin between the safe and restricted fields of travel. For the TSR, a safety margin of 1.0 nm is chosen.

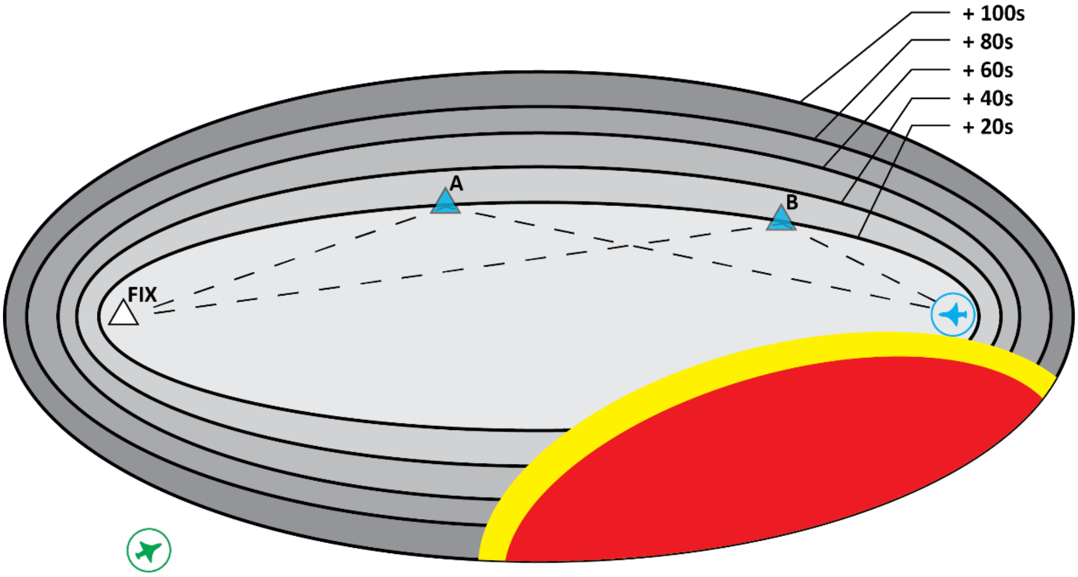


Figure 2: Example of TSR. The grey area resembles the safe field of travel. The red area shows the restricted field of travel. The TSR outer ellipse is limited by the maximum allowable delay, which is 100s in this example. The different shades of grey give an indication in terms of the optimality of that area in terms of time delay. Routes A and B have equal track length and delay.

Additional Interface Elements

In the experiment you will be provided with additional layers of information, next to the TSR, that gives insight in how the automation makes its decision. The visual information is based on the crucial elements of automation. It focuses on how automation is:

1. Discretizing the problem
2. Selecting points for evaluation
3. Determining feasibility of evaluated cells
4. Determining the optimality of feasible cells

In this briefing, the design will not be discussed in detail. During the training phase of the experiment, you will be provided with a thorough explanation of the interface itself and the information it is portraying.

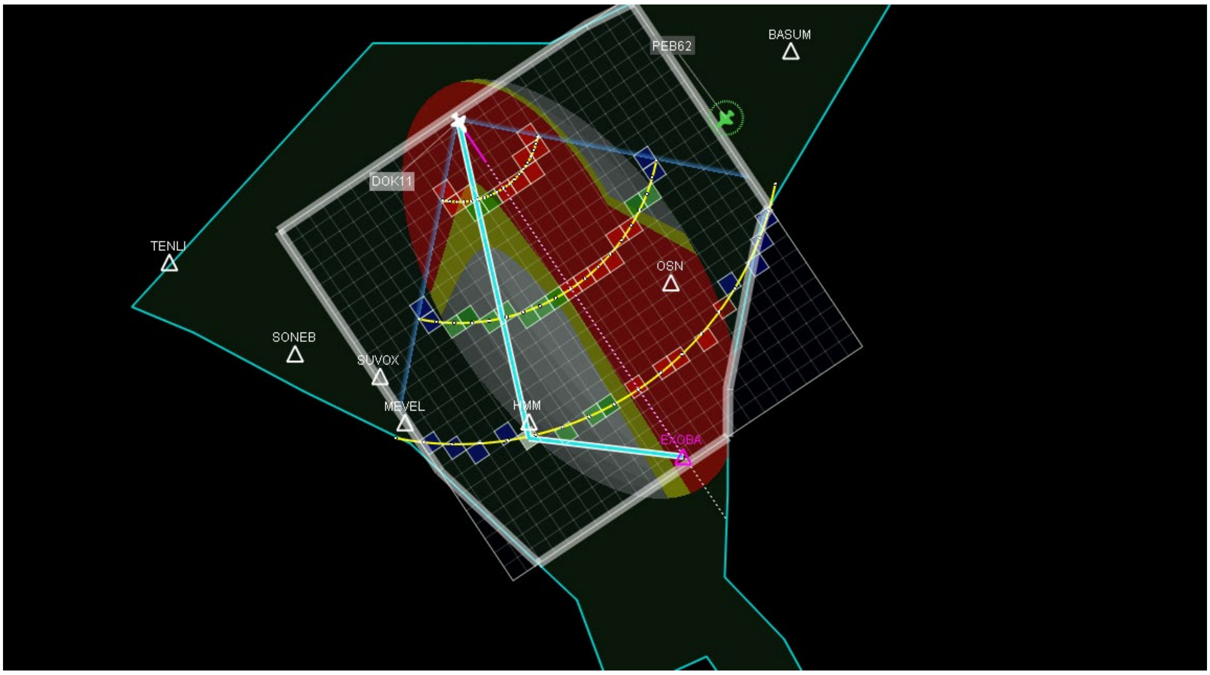


Figure 3: Interface at full transparency. Detailed information will follow in the experiment session.

Information shown is dependent on the level of transparency. At higher levels of transparency, more information will be given. In total, there are 4 levels. These levels will be discussed during training. Figure 3 gives an indication of what the interface looks like at maximum transparency level. In Figure 4, the transparency slider can be found. During the experiment it is not possible to adjust the slider. However, it can be used to check the current level of transparency.

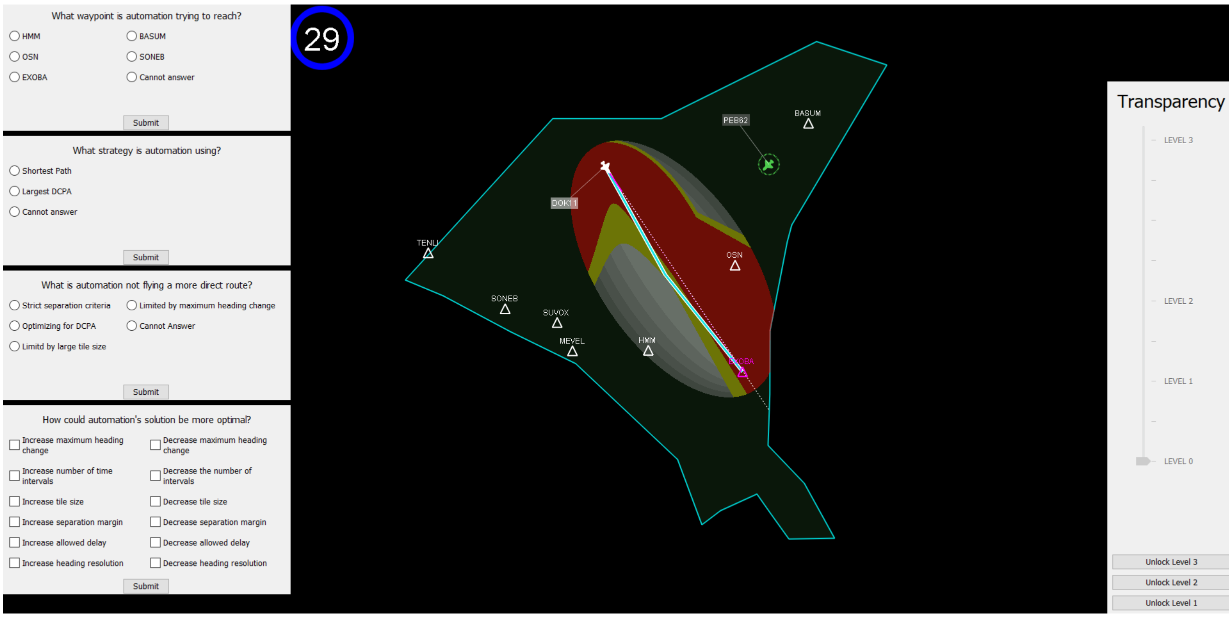
Figure 4: Transparency Slider

3. Experiment Setup & Goal:

The experiment is held at the Faculty of Aerospace Engineering of the TU Delft. It will take up to 2 hours. The traffic scenarios will be presented on a display and interactions with the simulation are done through mouse inputs.

During the experiment, you will be given a large variety of static traffic situations. In these situations, rerouting is necessary and automation has already proposed a new route. Your goal in the experiment is to answer questions about the choice of automation to the best of your ability. There is a time limit for each scenario. However, these are set such that it should be no problem to finish the scenario within the time limit given to it. You will be warned in time if you are getting to close to the time limit. Before the experiment is performed, you will be explained all features of the interface. You will have the opportunity to train with the interface and questions before the real experiment starts.

Each scenario has 4 questions. You are asked to answer these questions at 4 different levels of transparency. Once all questions are answered at a level, the transparency increases and you are able to change your answers if you come to new insights due to new information at this higher level of transparency. Additionally, you will be asked how confident you are about your answers.



What waypoint is automation trying to reach?

○ HMM ○ BASUM
○ OSN ○ SONEB
○ EXOBA ○ Cannot answer

What strategy is automation using?

○ Shortest Path
○ Largest DCPA
○ Cannot answer

What is automation not flying a more direct route?

○ Strict separation criteria ○ Limited by maximum heading change
○ Optimizing for DCPA ○ Cannot Answer
○ Limitd by large tile size

How could automation's solution be more optimal?

☐ Increase maximum heading change ☐ Decrease maximum heading change
☐ Increase number of time intervals ☐ Decrease the number of intervals
☐ Increase tile size ☐ Decrease tile size
☐ Increase separation margin ☐ Decrease separation margin
☐ Increase allowed delay ☐ Decrease allowed delay
☐ Increase heading resolution ☐ Decrease heading resolution

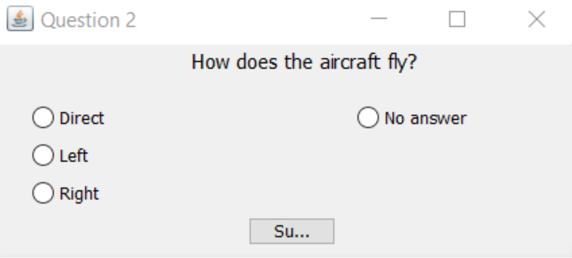
Figure 5: Overview of display during experiment

Figure 5 gives an overview of what the display will look like during the experiment. The airspace sector is the same throughout the experiment. In the current situation, two aircraft fly across the sector. One aircraft is selected and automation has proposed a new route. To the right the transparency slider, can be seen. On the left, 4 questions are popped-up. These questions needs to be answered. Next to the questions, a small clock will pop-up when you only have 30 seconds left to answer the questions of that level of transparency.

During the experiment:

1. Answer all question to the left, to the best of your ability. As quickly as possible [but don't feel rushed].
2. Once you have answered a question, indicate how confident you are about that answer on the confidence slider.
3. Please do not feel obligated to answer the questions from top to bottom. You are not bound by the order in which they are given.
4. Be aware of the three types of questions:
    a. Radio button question: you are only able to select one answer. See example in figure 6.
    b. Check box question: you are able to select multiple answers. See example in figure 7.
    c. Confidence question: you can indicate on a slider how confident you are about your answer. These questions always pop-up after answering one of the above mentioned questions.
5. There is always a "Cannot answer" option in the question. Please make use of this option when you think you cannot answer the current question with the information provided to you.
6. Once all questions are answered, the next transparency level will be given including the same questions as before. Please check if you would like to change your answer or confidence level. There are 4 levels.
7. 30 seconds before the time limit, a clock will appear warning you to answer on time. The time limit per level is 1.5min except for Level 0. At Level 0 the time limit is 3min.
8. If you have answered all questions at the highest level of transparency, the next scenario will be loaded directly.
9. Once you press "start next scenario" the next scenario will be started.

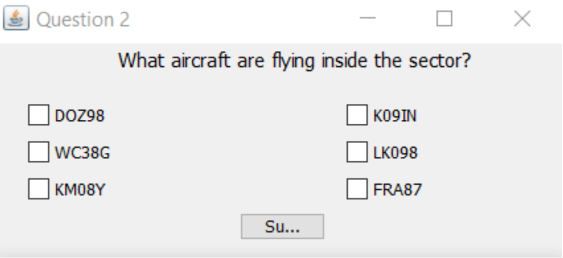
The final part of the experiment is the debriefing. In this phase, you are asked to fill in a questionnaire. There is plenty of opportunity in this phase to leave any comments that you think are relevant. All feedback on how the interface and experiment could be improved is very welcome!



How does the aircraft fly?

○ Direct
○ Left
○ Right
○ No answer

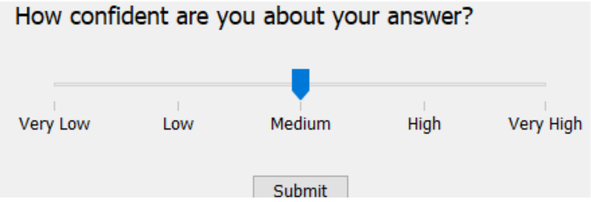
Figure 6: Radio Button Question



What aircraft are flying inside the sector?

☐ DOZ98
☐ WC38G
☐ KM08Y
☐ K09IN
☐ LK098
☐ FRA87

Figure 7: Check Box Question



How confident are you about your answer?

Figure 8: Confidence question

The overall planning of the experiment is shown in table 1. It is expected that the experiment will take about 3.5hrs.

Schedule	
Introduction to the experiment	10 min
Fill in questionnaire & Informed Consent	10 min
Interface explanation and training	60 min
Break	10 min
Measurement round 1	40 min
Break	10min
Measurement round 2	40 min
Fill in debriefing questionnaire	20 min
Short debrief	10 min
Total	***3.5hrs***

D

Code Overview

This chapter gives an overview of the software developed during this master thesis. It is divided into three parts. In Section D.1 an overview of all code files is provided. A flow diagram for the visualisation of the Flexibility Metric is shown in Section D.2. An overview of the settings is given in Section D.3.

D.1. Files

An overview of the most important files used in the code can be found in Figure D.1. The files are grouped into two main groups. One group gives an overview of the files that together form the Flexibility Metric. These have not been created during this research. However, because they form an important starting point for the visualisation, they have been included in this overview as well and are further discussed in Section D.1.1. The second group contains files that are used in order to visualize the Flexibility Metric and/or are important parts of the experiment environment. They are discussed in Section D.1.2 and Section D.1.3 respectively.

D.1.1. Flexibility Metric

A detailed explanation about how the Flexibility Metric itself works can be found in Section A.3. This section relates the concepts discussed in Section A.3 to the code of SectorX. First, the core infrastructure of the the Flexibility Metric is discussed. Subsequently, the strategies of the Flexibility Metric are explained.

D.1.1.1 Core Elements

If a new trajectory is requested, a new thread is created in which the calculations for a trajectory are performed. Therefore, it does not limit the performance of the other threads of the SectorX simulation. In this new thread, the file *TrajectorGenerator.java* is called. This function will produce a new trajectory for the aircraft to a target waypoint. The current state of the aircraft and the states and intents of other aircraft are taken into account for creating a new trajectory. Furthermore, settings for the trajectory generator are stated in the config file as will be further discussed in Section D.3

The algorithm starts by creating a MetricGrid as is defined in *MetricGrid.java*. This grid contains information on all constraints that the algorithms is considering. Important elements of this grid are the obstacle strategy, interval strategy, state change strategy and grid strategy.

A MetricGrid is created for several delay steps. Furthermore, for each interval percentage and the number of intermediate waypoints a new MetricGrid is created. During this research, this was simplified by limiting the number of intermediate waypoints to only one waypoint (Dog Leg). Therefore, a Metric is created for each interval strategy. In each MetricGrid, the algorithm tries to find a solution. All MetricGrids are saved in the object MetricGridMap.

The MetricGrid has a TimeCellMap for each interval of the interval strategy. The TimeCellMap contains TimeCells, which size and shape are determined in the grid strategy. During Forward Propagation it is investigated which of these TimeCells are reachable considering the constraints, such as those defined in the state change strategy. Branches are created to TimeCells that are within reach

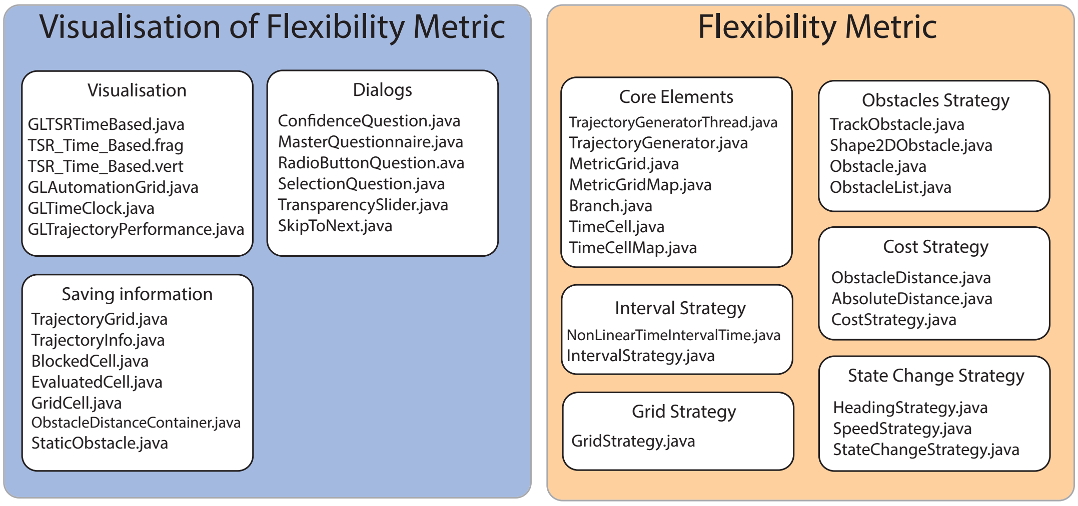


Figure D.1: Files used in SectorX. A distinction is made that between files of the Flexibility Metric and the files used for the visualisation and usage of the interface elements.

of each other. A Branch and TimeCell are not created if it would lead to a conflict. During backward propagation, all branches and TimeCells are set feasible that result in a trajectory from start to end point that full-fill all constraints.

D.1.1.2 Grid Strategy

The grid strategy is defined in *GridStrategy.java*. The grid strategy takes as inputs the starting point, end point and the size of the cells. This information is used to create a grid that forms the basis of the MetricGrid. All cells that are created have been given IDs, which are used throughout the codes. Built-in functions make it easy to switch from grid-ID to vector locations.

D.1.1.3 Interval Strategy

Multiple methods for creating time intervals have been created. In this research, *NonLinearTimeIntervalTime.java* is used. This method is able to work with delay and can create intervals at particular percentages of travel time. As stated before, multiple time intervals are created, because an interval strategy is created for each delay step and each interval percentage.

D.1.1.4 Obstacle Strategy

Obstacles are saved in *ObstacleList.java*. Both static and dynamic obstacles are saved in this list. Static obstacles objects are defined in *Shape2DObstacle.java* and dynamic objects are defined in *TrackObstacle.java*. This information is used in forward propagation to check for conflicts.

D.1.1.5 State Change Strategy

The State Change Strategy contains two important strategies that are defined in *HeadingStrategy.java* and *SpeedStrategy.java*. The Heading Strategy consists of both the maximum allowed heading change and the heading increments. The Speed Strategy has a constant speed in this research. It is possible to use a dynamic speed strategy. If that is desired, a maximum speed, a minimum speed and speed increments should be defined.

D.1.1.6 Cost Strategy

A cost function is used for selecting the optimal trajectory. Two cost metrics have been used in this research. *ObstacleDistance.java* assigns costs to cells according to their closest distance to an obstacle. *AbsoluteDistance.java* assigns cost to cells according to their total track length.

D.1.2. Visualisation

An important contribution of this research was creating interface elements that visualizes the inner-working of the Flexibility Metric algorithm. The following subsections describe the core files that are used for this purpose.

D.1.2.1 Saving information

The TrajectoryGenerator only has a trajectory as output. This information is a vital part of the decision, but it does not give a complete image of why exactly that decision has been made. A lot of information in the software is not saved anywhere and therefore could not be used for visualisation. Therefore, the first step was to create an infrastructure that saved the necessary information. For that purpose, the *TrajectoryInfo.java* file was created. This file saved all relevant information of the algorithm such that this information could be visualised.

The following information is saved in the TrajectoryInfo object:

1. Trajectory grid
    - Track distance
    - Closest distance to obstacle
    - Cell status
    - Costs
    - Obstacle IDs
2. Grid strategy
3. Time intervals
4. State Change Next States

Note that the the Trajectory Grid is an object from *TrajectoryGrid.java*. This object saves information such as their status or assigned costs, in GridCells. If the GridCell has the “Blocked” status, the ID of the obstacle which triggers this status is saved.

Important to understand is that the TrajectoryGrid is not the same as a MetricGrid or TimeCellMap. For each trajectory proposal there will always be only one TrajectoryGrid, while there could be thousands of MetricGrids and TimeCellMaps. This is because for each delay step and interval percentage new MetricGrids and TimeCellMaps are created. The TrajectoryGrid processes all of this information in order to build a grid that can be used for visualisation.

D.1.2.2 GLAutomationGrid

Non-interactive visualisation of the Flexibility Metric can be found in *GLAutomationGrid.java*. This file draws the proposed trajectory of automation. Additionally, the grid and the outer edges of the evaluated area are created in this file. Furthermore, the status of the cells is made clear by colouring the cells. The optimality of feasible cells is indicated using different shades of green. Also the state change bands and dots are created in this class. Finally, the heading lines can be found in this file as well.

D.1.2.3 GLTrajectoryPerformance

Interactive interface elements can be found in *GLTrajectoryPerformance.java*. Functions in this file ensure that a label pops-up while hovering over cells. Depending on the transparency levels and type of cell, additional information is shown. GLTrajectoryPerformance draws in the label the status of the cell. In case the cell is a blocked cell, hovering over the cell triggers highlighting of the reason of that blockage (another aircraft or restricted airspace). Hovering over feasible cells shows more information about the cost and information that is used for creating the cost. This information can be either the extra distance flown compared to the direct path or the closest distance to an obstacle at any point of the trajectory.

D.1.2.4 GLTSRTimeBased

The TSR is created using a shader. The interaction between the java code of SectorX and the shader program can be found in *GLTSRTimeBased.java*. A variety of settings regarding the TSR can be changed in this file such as colors, separation margin, maximum delay and many more. The shader program contains the physical relationship that determine the regions of the TSR. The shader program consists of two files $TSR\_Time\_Based.frag$ and $TSR\_Time\_Based.vert$. The *.vert* file is a simple code defining the position of world and screen. The *.frag* file contains the actual dynamics.

D.1.2.5 TransparencySlider

A dialog was implemeted that contained a slider which could be used to adjust the transparency level. The dialog can be found in *TransparencySlider.java*. Note that the slider was only used during training to explain all interface elements. The sliders value determines weather a interface elements is visualised. This done using booleans and thresholds. When a certain threshold is reached, a boolean will be switched to true making it visible. These thresholds are predefined and differ per transparency direction. In order to facilitate a smooth transition through the levels, an alpha (transparency) value is used to slowly fade-in and fade-out interface elements.

D.1.3. Experiment

In order to test the interface, extra elements were created. In the experiment, participant were required to answer questions about automation. These questions needed to be answered within a certain time limit and participants were notified 30 seconds in advance if they were running out of time. Finally, a skip button was created in order to easily skip through the experiment for testing scenarios. All three elements will be discussed in more detail in the following sections.

D.1.3.1 Questions

Questions were either single or multiple answer. *RadioButtonQuestion.java* is a dialog that is used for single answer questions. Multipe answer question are created using *SelectionQuestion.java*. Once a question has been answered a confidence slider appears to ask about their confidence in their answer. The confidence slider can be found in *ConfidenceQuestion.java*.

Questions differ slightly between scenarios. Therefore, the questions and answer options could be changed for each scenario and each question position. In the Config file of a scenario, the question-ID can be indicated per question position. Questions and their answer options can be created in *MasterQuestionnaire.java* by adding a new element, with the question-ID as key, to the hashmap for questions. The answer options can be added in a similar manner to the hashmap for answer options.

D.1.3.2 Clock

In order to indicate that the participants had only 30 seconds left, a clock was created. This clock can be found in *GLTimeClock.java*. The clock is only shown if the time exceeds a thresholds. This thresholds increases per transparency level. By doing this, a time per transparency level can be set. In the experiment, the participants had 3 minutes to complete the Level 0 and 1.5 minutes for each of the other levels.

D.1.3.3 Skip Button

When designing scenarios and playlists, it is handy to be able to skip through scenarios. Therefore, a skip button was created that ended the current scenario and loaded the next one. The skip button can be found in *SkipToNext.java*.

D.2. Flow Diagram Visualisation of Flexibility Metric

Figure D.2 contains a flow diagram of the visualisation of the Flexibility Metric. This flow diagrams is meant to give a better overview of the information presented in the previous section. The diagram is divided into two parts. One part focuses on the Flexibility Metric itself and how it produces a trajectory. A more detailed overview of this Flexibility Metric is given in Section A.3.

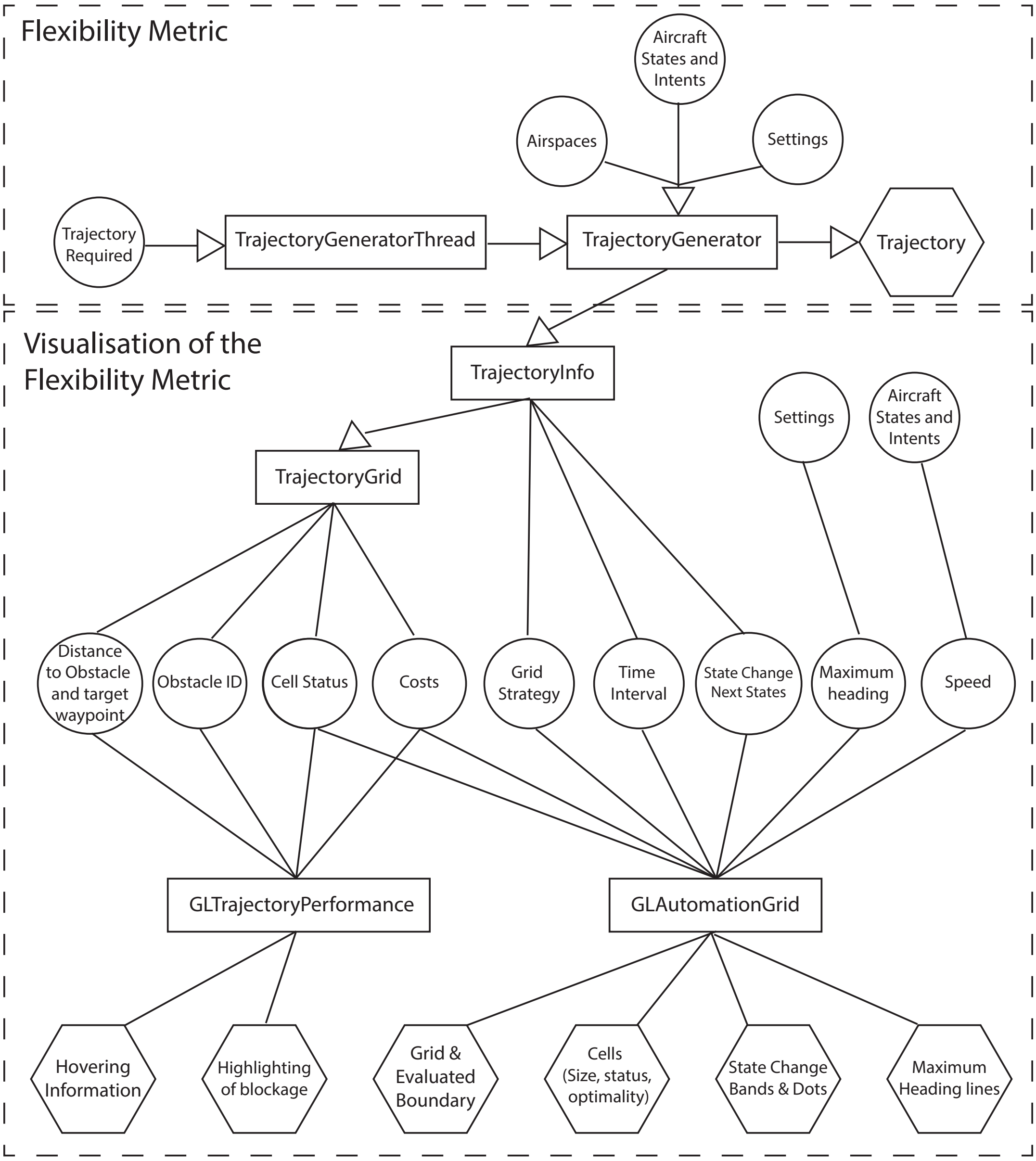


Figure D.2: Flow diagram of code. Blocks are code blocks. Circles represent information. Hexagon are the outcomes.

The second part of the flow diagram highlights the important files and information for the visualisation of the Flexibility Metric. Note that information is extracted from the TrajectoryGenerator by TrajectoryInfo and post-processed by it in order to make information ready for visualisation. The two main files that create the visualisation are the GLTrajectoryPerformance and GLAutomationGrid. The resulting interface elements are noted in hexagons at the bottom of the flow diagram.

D.3. Config and Playlist

New parameters have been added in the config file of SectorX. An overview of the adjustable parameters can be found in Table D.1. The names used in the SectorX environment are presented together with the default value of these parameters. A description of each parameters has been provided as well. During the experiment, settings changed in order to develop interesting and non-trivial scenarios.

The playlist used in the experiment can be found in Table D.2. The playlist trigger scenario and config files. By investigating the playlist, the corresponding scenario and config files can be found easily. Additionally, a playlist for training and for making screenshots of the interface are indicated in the table.

Table D.1: Config settings that were added to the SectorX code during the development of the additional interface elements.

Name	**Default**	**Description**
SHOW_TSR	TRUE	Show TSR based on time
TRA_GEN_USE_TILE_SIZE	TRUE	Use tile size instead of number of tiles
TRANSPARENCY_DIRECTION	TRUE	Use bottom-up (TRUE) or Top-down (False)
TRA_GEN_BOUNDBYSECTOR	TRUE	Color the evaluated area boundary
TRA_GEN_SHORTESTPATH	TRUE	Optimizing for shortest path
TRA_GEN_OBSTACLEDISTANCE	TRUE	Optimizing for largest separation margin
TRA_GEN_ACTIVATE_UP_ON_START	TRUE	Generate trajectory when scenario is loaded
SHOW_TRA_GEN_DECODE	FALSE	Show extra (debug) information
SHOW_TRA_GEN_KEEP_SELECTED	TRUE	Prevents the ability to select other aircraft
SHOW_TRA_GEN_USE_TRANSPARENCY_SLIDER	FALSE	Interactively determine the transparency level
SHOW_SKIP_DIALOG	TRUE	Show the skip dialog
SHOW_TRA_GEN_QUESTIONNAIRE	TRUE	Show the Questionnaire
QUEST_POS_1_TYPE	TRUE	Q1 single answer (TRUE) or multiple answer (FALSE)
QUEST_POS_2_TYPE	TRUE	Q2 single answer (TRUE) or multiple answer (FALSE)
QUEST_POS_3_TYPE	TRUE	Q3 single answer (TRUE) or multiple answer (FALSE)
QUEST_POS_4_TYPE	FALSE	Q4 single answer (TRUE) or multiple answer (FALSE)
TSR_MAX_DELAY	100	Maximum allowed delay of TSR [s]
AIRCRAFT_SEPARATION_AUTOMATION_MARGIN_NM	0.5	Margin taken by automation [NM]
TRA_GEN_NUM_WAYPOINTS	2	Number of segments used [-]
TRA_GEN_MAX_DELAY	100	Maximum allowed delay of Automation [s]
TRA_GEN_MAX_HEADING	45	Maximum allowed heading change [deg]
TRA_GEN_DELTA_HEADING	5	Heading increment [deg]
TRA_GEN_NUM_GRIDPOINTS	10	Number of tiles [-]
TRA_GEN_TILE_SIZE	4630	Size of a tile [m]
TRA_GEN_START_PERCENTAGE	20	Percentage for first time interval [%]
TRA_GEN_INCREMENT_PERCENTAGE	30	Percentage increment for time interval [%]
QUEST_POS_1	1	Q1 question ID
QUEST_POS_2	2	Q2 question ID
QUEST_POS_3	3	Q3 question ID
QUEST_POS_4	4	Q4 question ID

Table D.2: Playlist used.

Playlist	Used for	Description
pX_F	Experiment	5 Scenarios of participant X in Bottom-Up method
pX_D	Experiment	5 Scenarios of participant X in Top-Down method
trainingP0	Training	9 Training scenarios
white	Screenshots	Altered version with a white background for screenshots

E

Additional Experiment Data

This appendix contains additional data that was not included in the article. Notation of scenarios and question can be found in Table B.1 and B.2 respectively. Bottom-up is noted as ‘I’ and top-down as ‘II’. A description of the type of plots is given in the following paragraphs:

Confidence Plots Figures E.1, E.2, E.3 show for each answer that a participant has given whether that answer was correct and how confident they were in giving that answer. The plots are per participant per questions. Therefore, there are three figures (Q2, Q3, Q4) each containing ten plots (one for each participant). Each plot contains the results of the ten experiment conditions (five scenarios times two directions) and for each condition four bars are made (each for every transparency level). Note that generally, participants indicate they cannot answer questions or answer question incorrectly at the lower level of transparency.

Distribution of Answers Figures E.4, E.5 and E.6 show the answers given per scenario per level. Colours indicate if answers were correct, incorrect or “cannot answer”. The bar always have an height of ten, because each of the ten participants have given answers for all these experiment conditions. Note that the lower levels of transparency have more incorrect and “cannot answers” than the higher levels of transparency. Furthermore, Q4, Figure E.6, has a relative high amount of wrongly answered questions. Participants said that this question was found a bit ambiguous and that they would have preferred more what-if probing functionalities to properly answer this question. Furthermore, Q4 was a multiple-answer question and it was only considered correct if all options were correctly chosen.

Confidence and Hit Ratio Figure E.7 show the distribution of the averaged confidence and hit ratios. Averages were taken per participant per direction and thus each data point is an average over 5 scenarios. The significant relationships stated in the article can be seen in these plots as well. Note the confidence of “cannot answer” has been inverted.

Assumption Testing Confidence and Hit Ratio Figure E.8 and Figure E.9 show hit ratio and confidence similar to Figure E.7. An important assumption in the hit ratio is that the “Cannot answer” was counted as an incorrect answer. In order to test the effect of this assumption, hit ratio plots were created in which “Cannot answer” were counted 33% and 50% correct. Note that a similar distribution is found in all plots. An important assumption in the confidence plots is that the confidence is inverted for “Cannot answer” answers. Confidence plots were created in which the “Cannot answer” confidences were not inverted and in which “Cannot answer” confidences were only converted at Level 0 (because it is unlikely they knew the answer at this level).

Confidence versus Hit Ratio Figure E.10 plot the average confidence level and hit ratio to each other. These type of plots were also included in the article. However, only per direction. In these plot the average of the hit ratio and confidence was not taken over 5 scenarios (per direction), but over 10 scenarios (both directions).

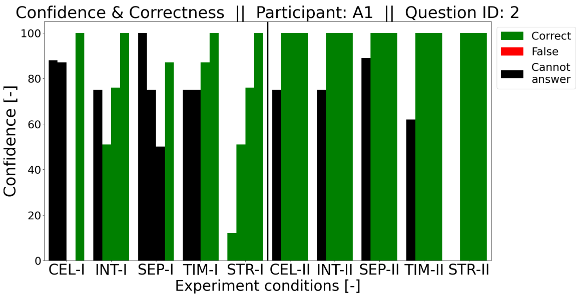
Table E.1: Order of question answering. Note that the participants primarily answered questions from top to bottom.

	First	Second	Third	Fourth
Q1	398	2	0	0
Q2	1	395	3	1
Q3	0	3	395	2
Q4	1	0	2	397

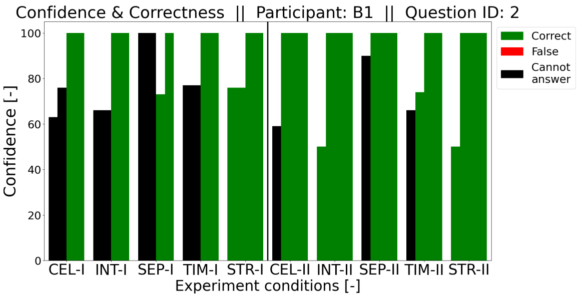
Answer Order Participants answered questions mainly from top to bottom. The frequency of questions answered in a particular order can be found in Table E.1. Note that Q1 is primarily answered first, Q2 secondly, etc. In total, rows and columns sum up to 400 responses, because each questions has been answered 400 times (4 levels, 10 scenarios, 10 participants).

Combined Time Plot of Group A and B Figure E.11 shows boxplots of the time required to complete the scenarios. A distinctions is made between group A and B. Lines connect data points belonging the same participant.

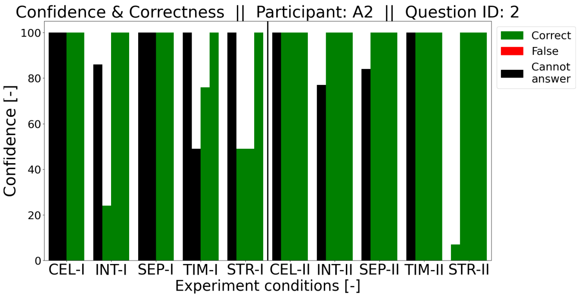
Interaction Figure E.12 contains plots considering the interaction with the interface. Because the threshold for counting something as a interaction was chosen arbitrary, plots with different thresholds are shown as well. Note that the other thresholds (0.1 and 1 seconds) both have a similar distribution of data points.



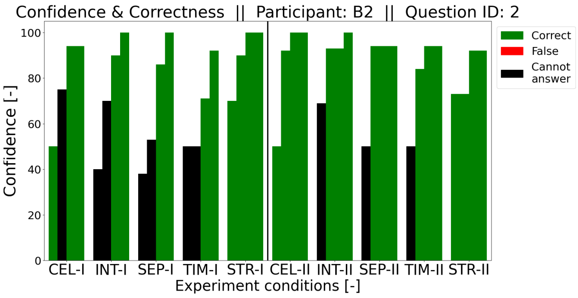
(a) Q2: A1



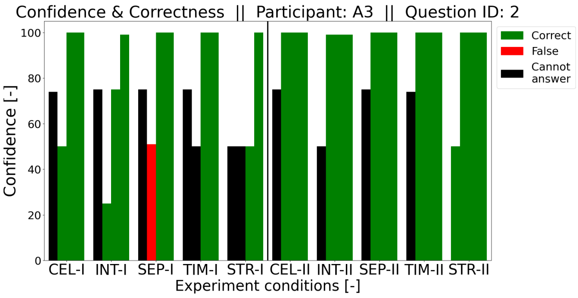
(b) Q2: B1



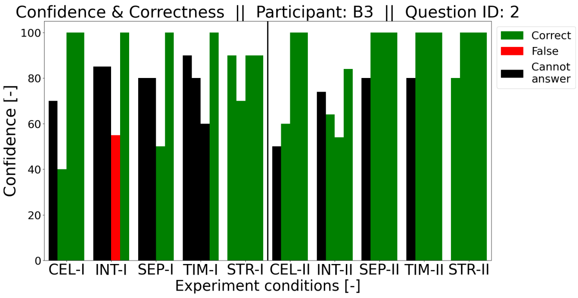
(c) Q2: A2



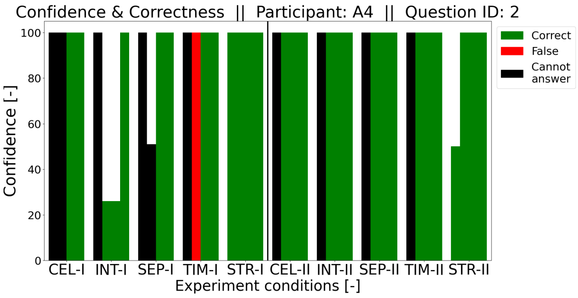
(d) Q2: B2



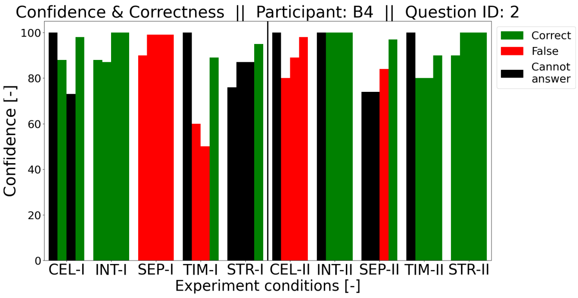
(e) Q2: A3



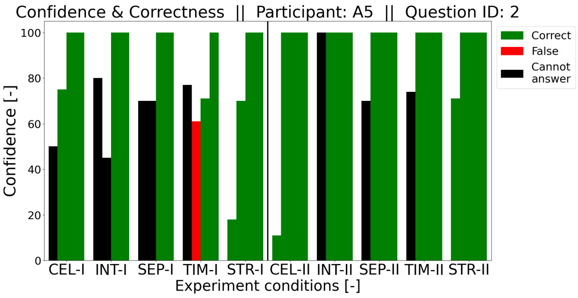
(f) Q2: B3



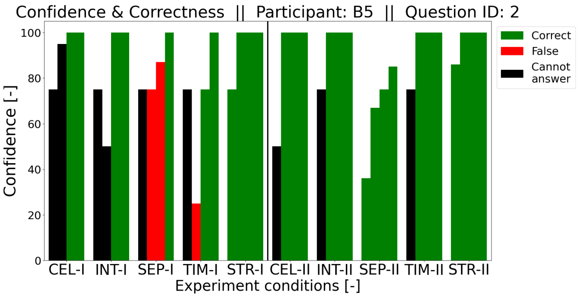
(g) Q2: A4



(h) Q2: B4

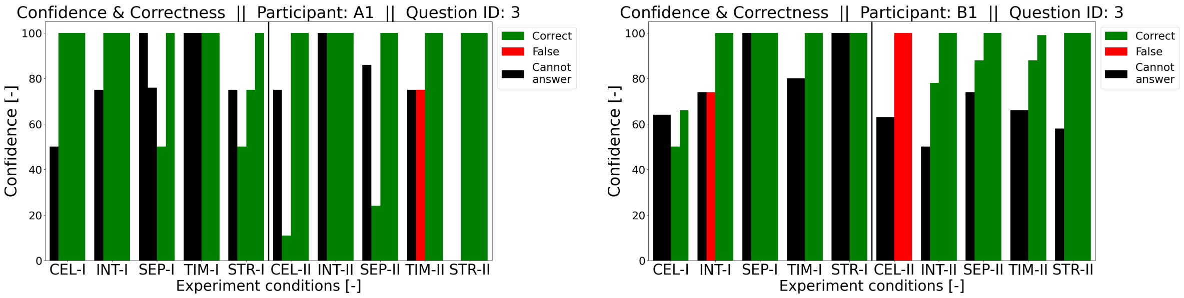


(i) Q2: A5

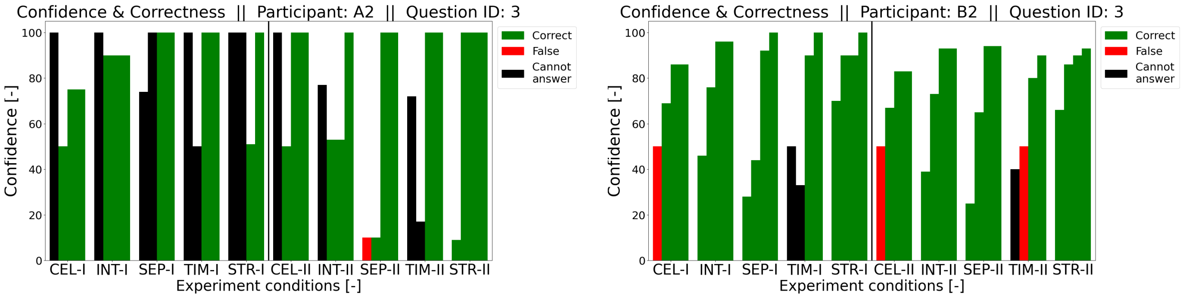


(j) Q2: B5

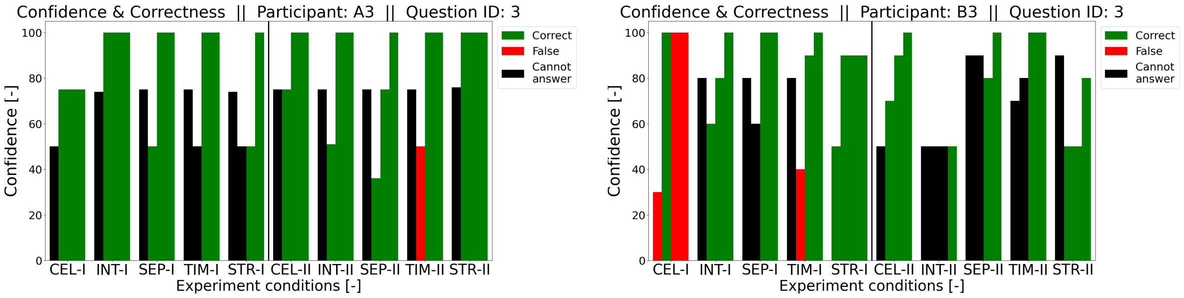
Figure E.1: Confidence per level per scenario for Q2 for all participants. Correctness, incorrectness and “Cannot answer” is indicated using color.



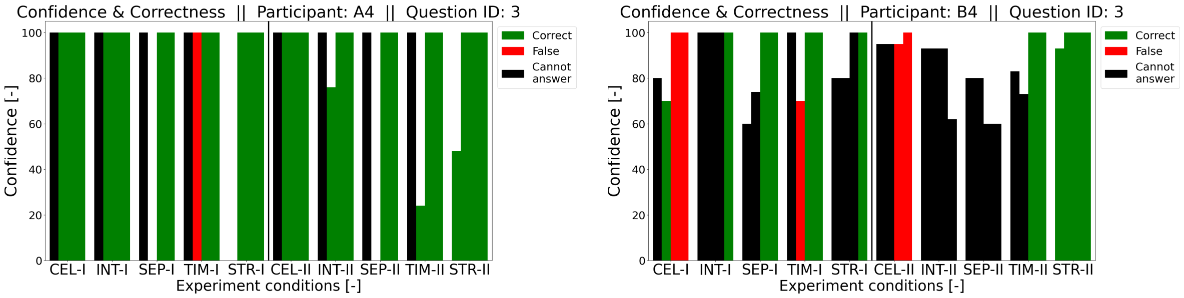
(a) Q3: A1 (b) Q3: B1



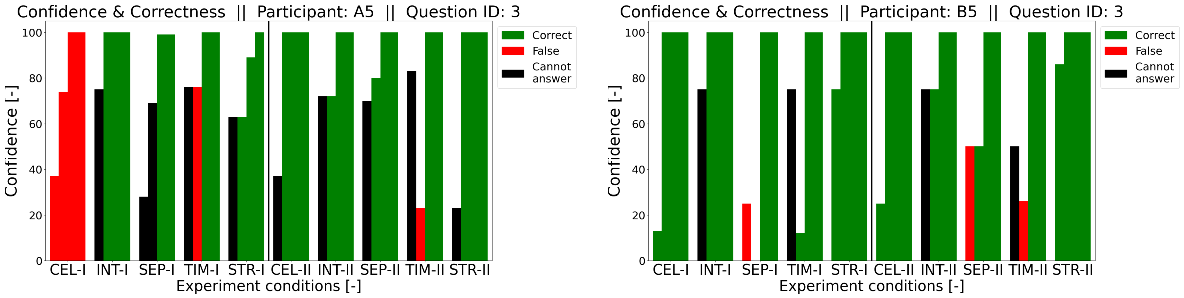
(c) Q3: A2 (d) Q3: B2



(e) Q3: A3 (f) Q3: B3

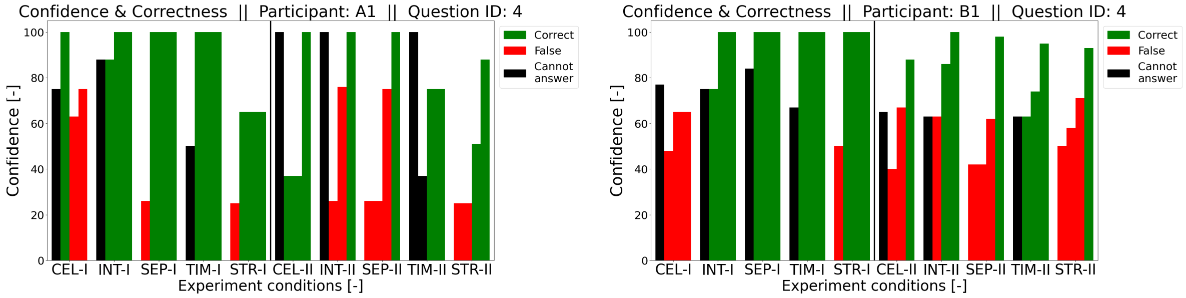


(g) Q3: A4 (h) Q3: B4



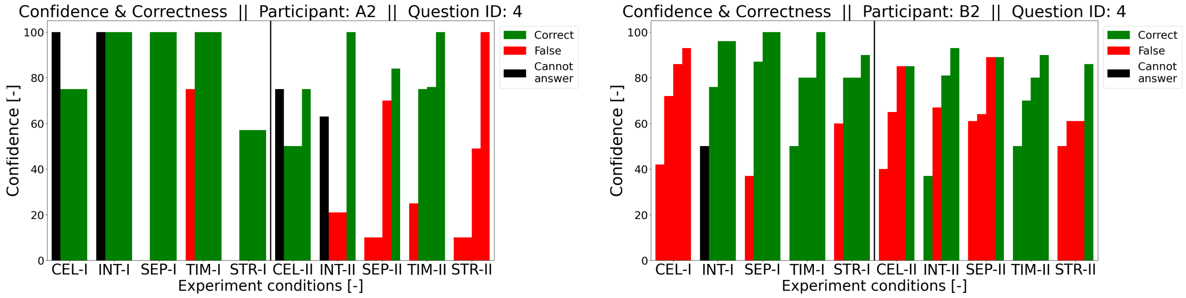
(i) Q3: A5 (j) Q3: B5

Figure E.2: Confidence per level per scenario for Q3 for all participants. Correctness, incorrectness and "Cannot answer" is indicated using color.



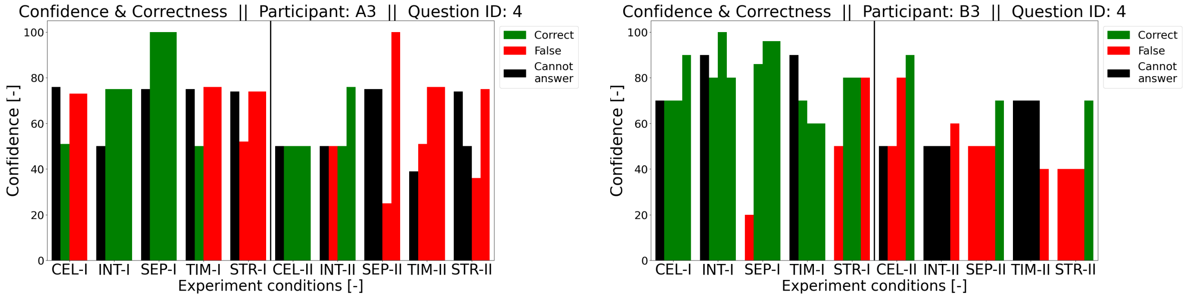
(a) Q4: A1

(b) Q4: B1



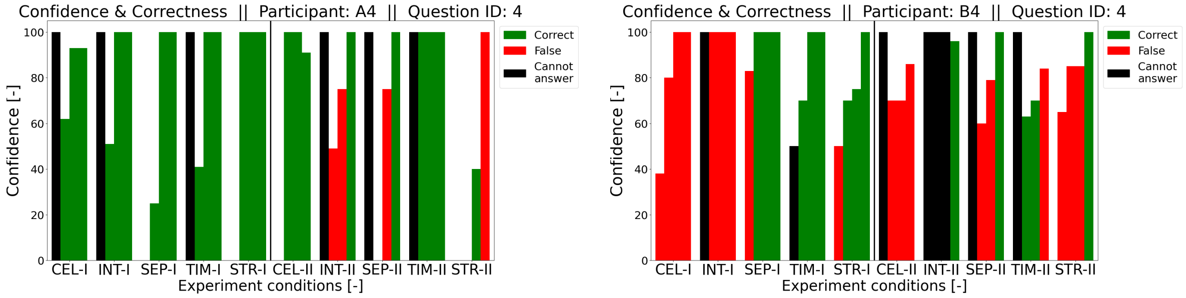
(c) Q4: A2

(d) Q4: B2



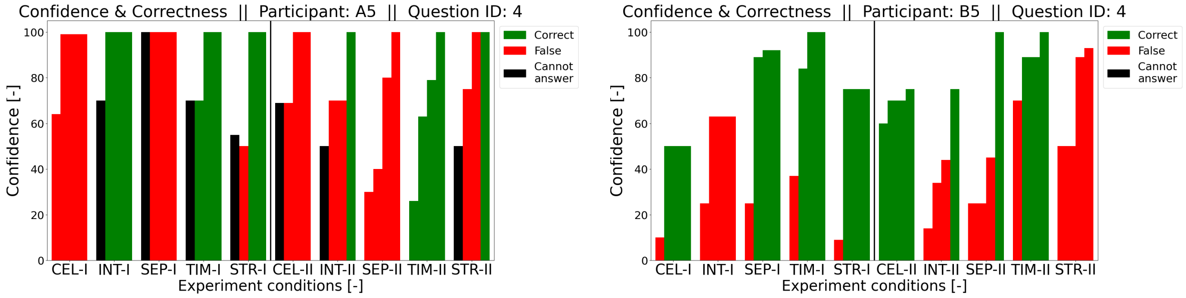
(e) Q4: A3

(f) Q4: B3



(g) Q4: A4

(h) Q4: B4



(i) Q4: A5

(j) Q4: B5

Figure E.3: Confidence per level per scenario for Q4 for all participants. Correctness, incorrectness and “Cannot answer” is indicated using color.

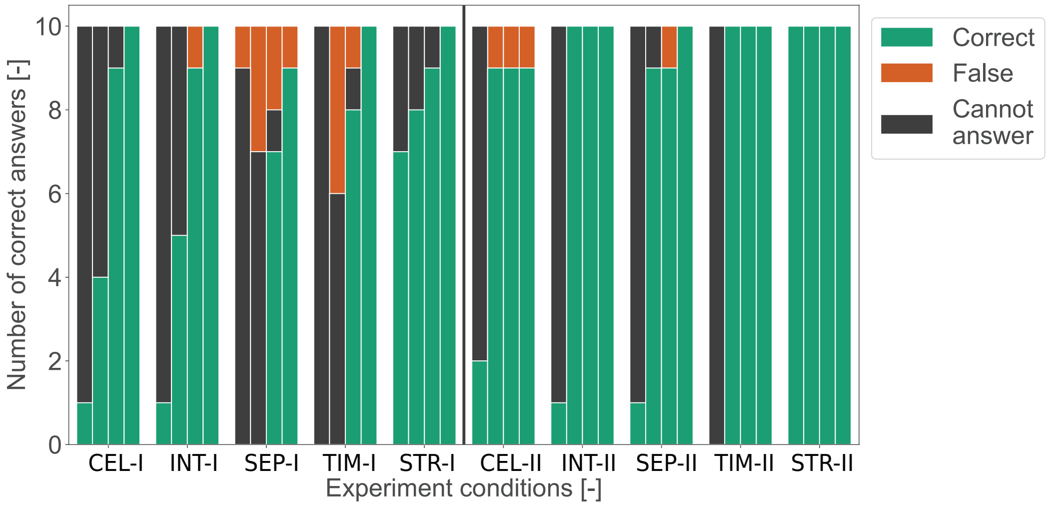


Figure E.4: The answers given per scenario per level for Q2.

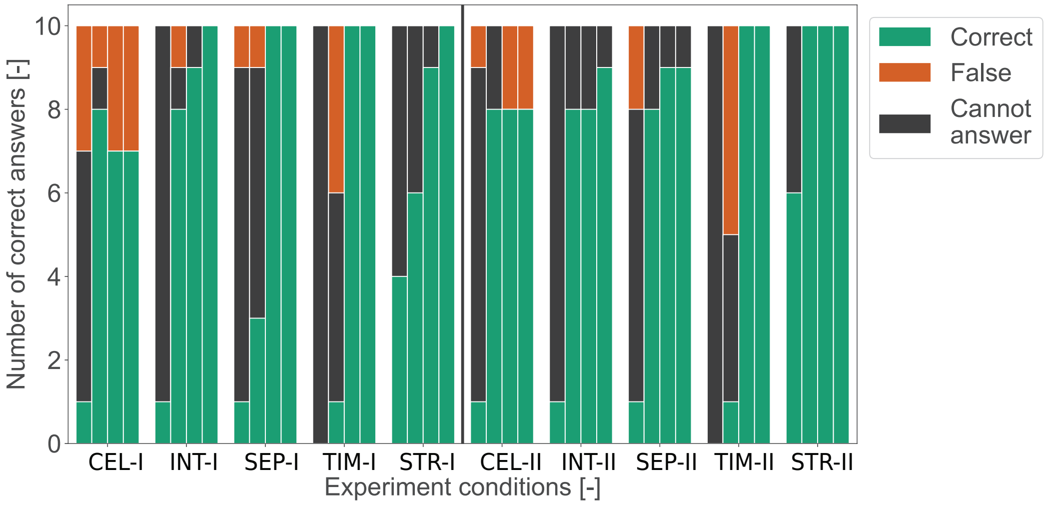


Figure E.5: The answers given per scenario per level for Q3.

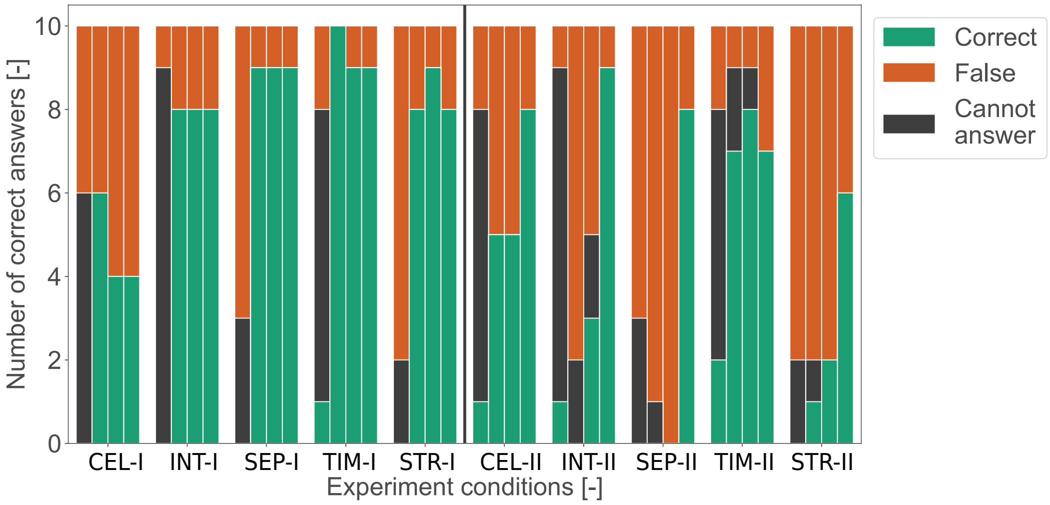
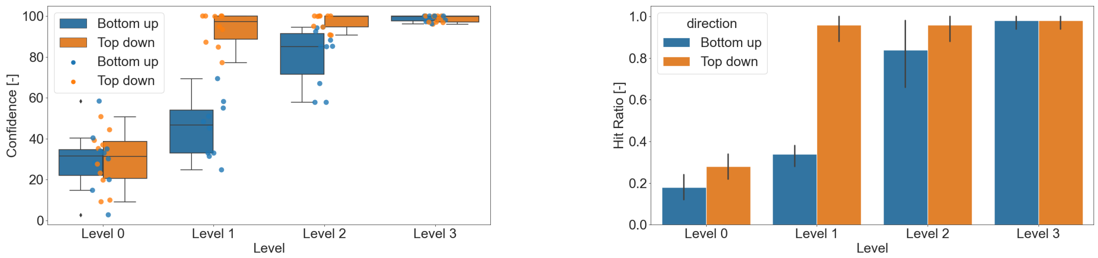
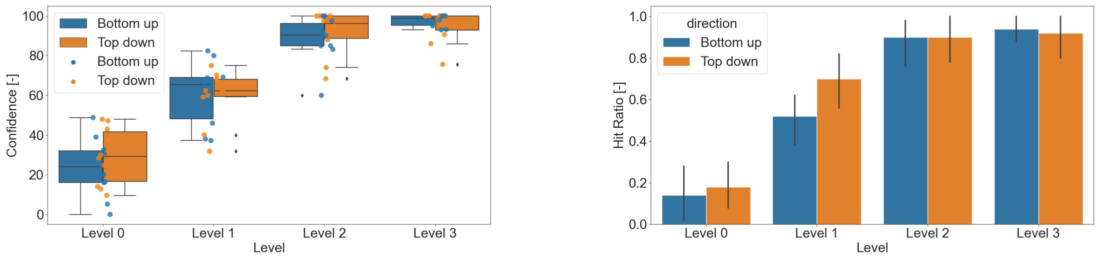


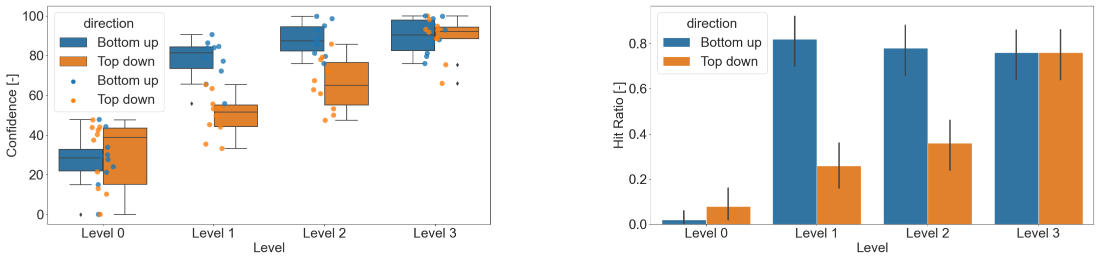
Figure E.6: The answers given per scenario per level for Q4.



(a) Q2: Confidence (b) Q2: Hit Ratio

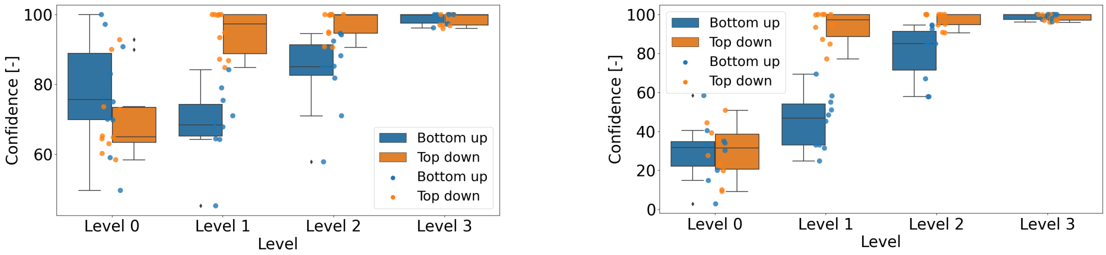


(c) Q3: Confidence (d) Q3: Hit Ratio

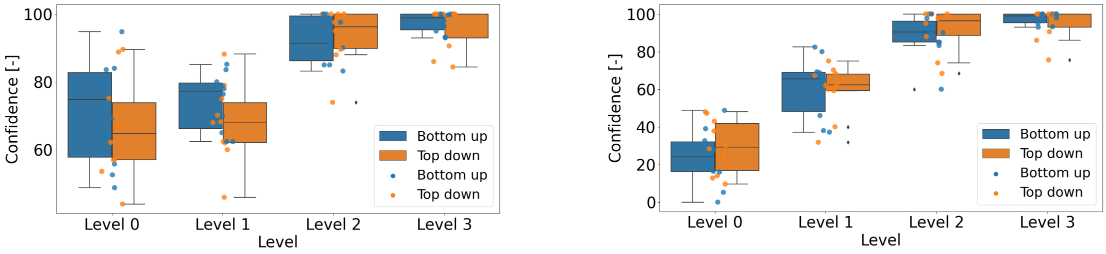


(e) Q4: Confidence (f) Q4: Hit Ratio

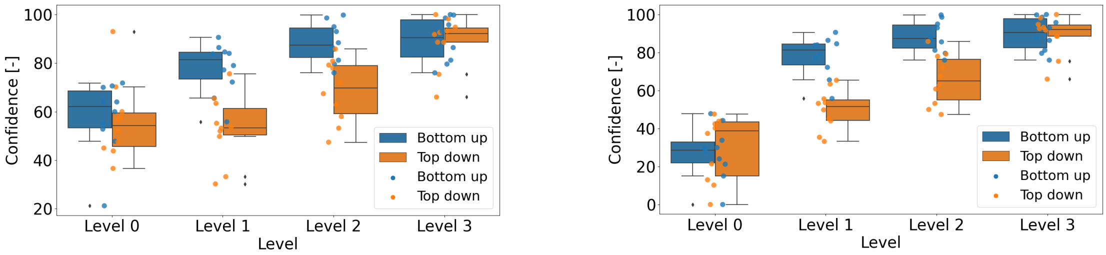
Figure E.7: Confidence and Hit Ratio for Q2, Q3 and Q4 for all levels. Bar plots have error bars of 95 percent. Each dot is a mean of one participant over 5 scenarios.



(a) Q2: Assumption A (b) Q2: Assumption B

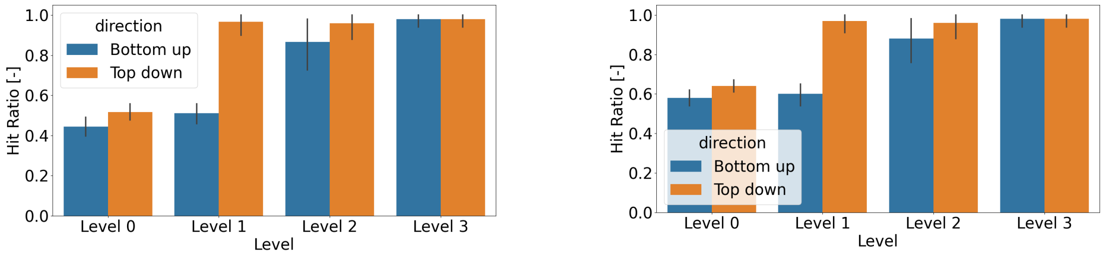


(c) Q3: Assumption A (d) Q3: Assumption B

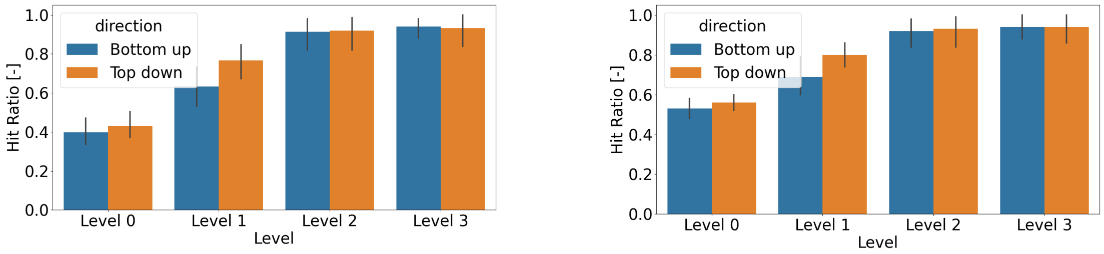


(e) Q4: Assumption A (f) Q4: Assumption B

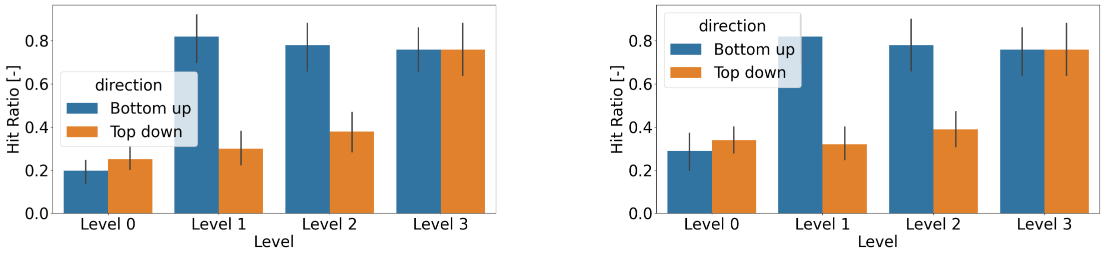
Figure E.8: Confidence for Q2, Q3 and Q4 for all levels for two assumptions. Assumption A does not invert confidence for “Cannot answer” and in Assumption B “Cannot answer” is only inverted at transparency level 0. Each dot is a mean of one participant over 5 scenarios.



(a) Q2: 33% correct (b) Q2: 50% correct

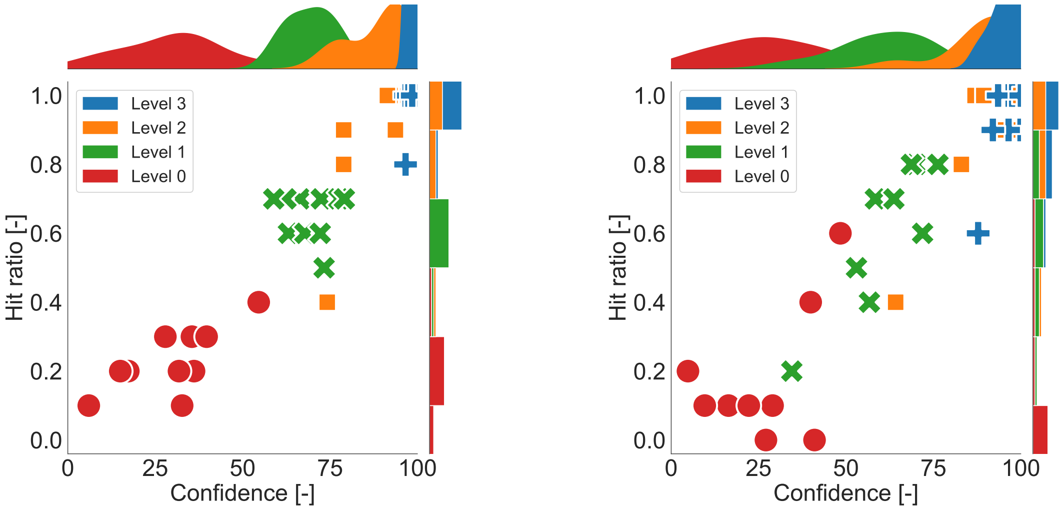


(c) Q3: 33% correct (d) Q3: 50% correct

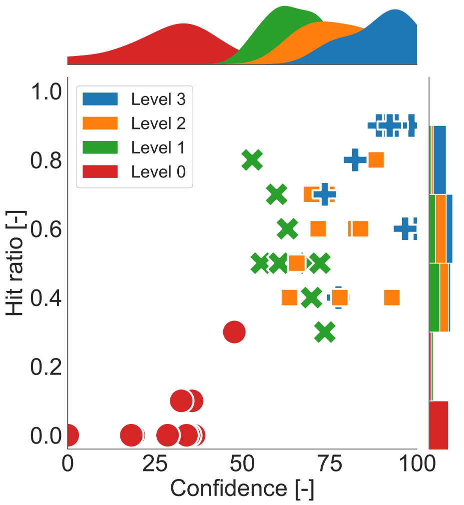


(e) Q4: 33% correct (f) Q4: 50% correct

Figure E.9: Hit Ratio for Q2, Q3 and Q4 for all levels for the assumptions that “Cannot answer” is 33% and 50% correct. Bar plots have error bars of 95 percent. Each dot is a mean of one participant over 5 scenarios.



(a) Q2 (b) Q3



(c) Q4

Figure E.10: Confidence versus hit ratio for Q2, Q3, Q4. Both top-down and bottom-up results are integrated in these plots.

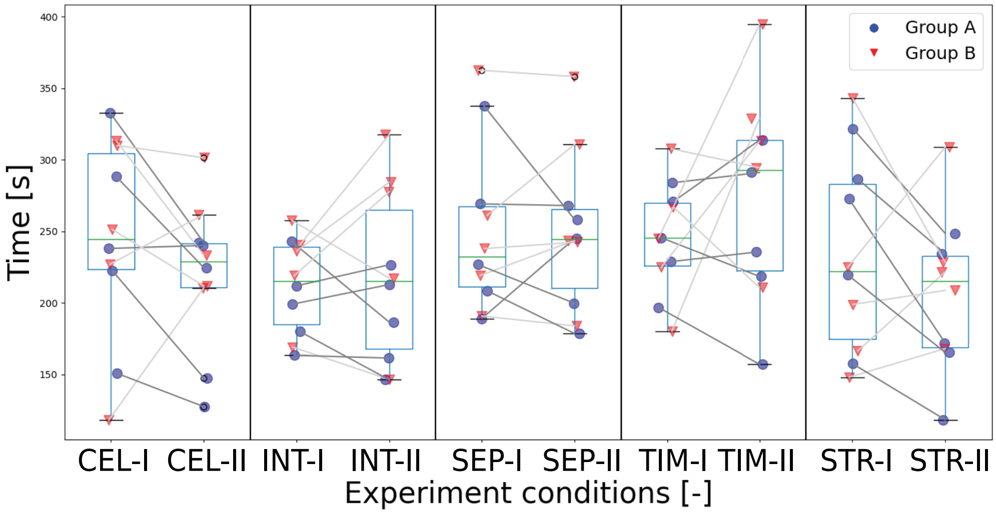
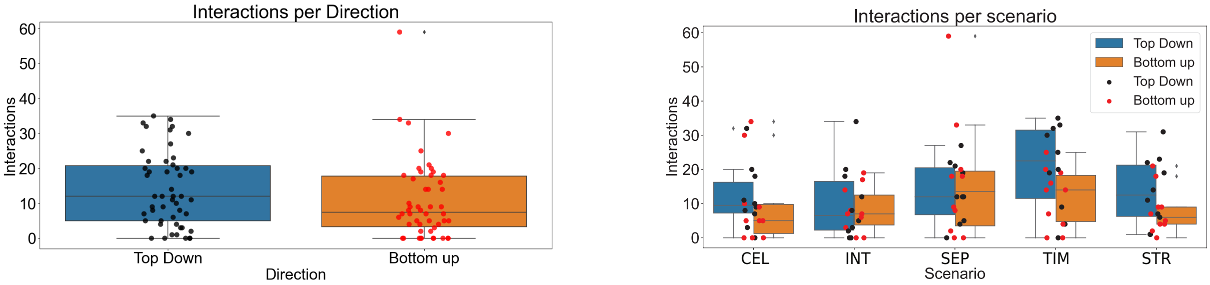
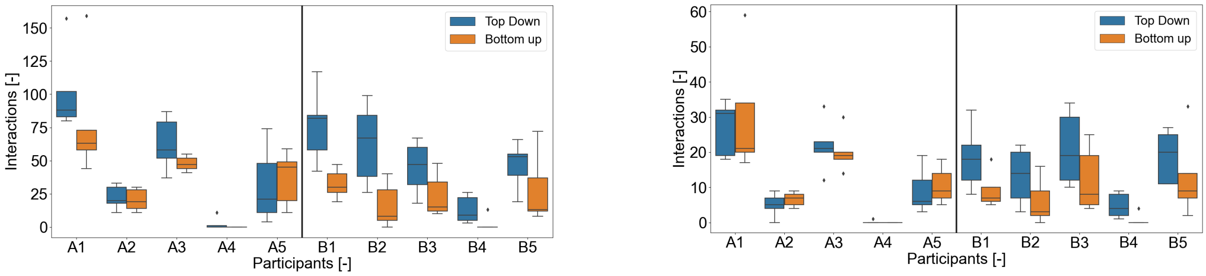


Figure E.11: Time per scenario. Participants of group A and B are indicated and the data points belonging to the same participant are connected through lines.



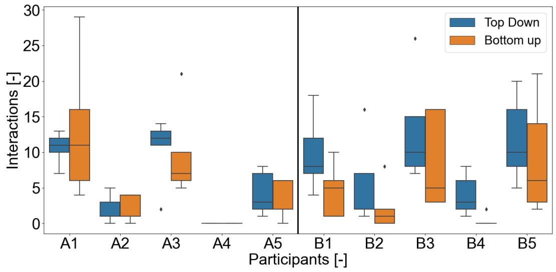
(a) Interactions per direction

(b) Interactions per scenario



(c) Interaction between participants. Thresholds = 0.1

(d) Interaction between participants. Thresholds = 0.5



(e) Interaction between participants. Thresholds = 1.0

Figure E.12: These plots show the interaction with the interface. Plot (a) shows the difference per direction (which is significant). Plot (b) shows the difference between interaction (also found significant but pairwise comparison yielded no significant differences between pairs). Plots (c), (d), (e) are the same plots, but with different thresholds for counting something as an interaction. Note that the distribution between all three remains about the same.

F

Post-Experiment Questionnaire Data

In this chapter the data from the post-experiment questionnaire are presented. The data was retrieved using the online Qualtrics software. The answers to the questions are shown using bar plots. Each bar plot has the corresponding in the caption. If there were any particularities, these are also indicated in the captions.

One question asked the participants to list the positive and negative attributes of the bottom-up and top-down approach. These are shown in Table F.1 and F.2 respectively.

Table F.1: Positive and negative comments on bottom-up approach by participants

Participant	Positive	Negative
A1	It is very clear what points the automation is considering and what should be changed to involve other points in the process	Visually, this method is already quite complex from level 1.
A2	Easier for finding mistakes in the heading increments, heading width and time interval settings	takes level 2-3 to provide good insight into why the automation chooses a specific route.
A3	Easy to follow the steps of the automation	Harder to know the strategy or constraints of the automation
A4	builds principle from the ground: really shows how the computer gets there in a natural order	slow
A4	-	requires higher levels to know more
A5	Immediately understand the discretization to understand the algorithms settings	A lot of information, not necessary of algorithm is known
A5	-	most useful (decision making) information only at level 3
B1	Questions 3 and 4 can already be partially answered with level 1	It takes an extra level of transparency (level 2) to get information about the time AND separation constraint. This information is particularly helpful to answer question 2 which I believe is slightly more important for an ATC before starting to evaluate different possible waypoints
B2	it quickly shows all the evaluated cells and why they were evaluated or exempted	it adds a lot of information to the screen quickly
B3	gradual, guided understanding of the situation; state change band useful to have early	not always sure about the status of the evaluated cells. this could have helped to decide why some points are not considered at an earlier stage [Blocked, timeout feasible]
B4	More clear boundary of the solution space at lower levels	You don't get to know from the start the useful for answering the question optimality of the cells which was
B4	It is useful to know the time interval (especially for answering these type of question) it is better to get this information earlier than later	-
B5	Fast clear picture of available headings	Most relevant information (feasible/selectable cells) only very high on transparency scale
B5	Fast view on considered cells	more Clutter

Table F.2: Positive and negative comments on top-down approach by participants

Participant	Positive	Negative
A1	Information at lower levels more tailored towards the needs of the controller	It is more easy to forget certain elements in lower transparency levels.
A2	upfront about the strategy used in the automation (level 1)	Q4 was difficult to answer at lower levels. Most of the time I answered these confidently at level 3
A3	Easy to find the strategy	Difficult to see the boundaries that the automation used and why certain points were not considered
A4	Most information can be gained from l1: good estimate is already made then	Can lead to false conclusions by trying to be too fast
A5	immediatly aware of the decision making	less aware about reasons for discretisation errors
A5	clear interface, less distractions	-
B1	Having the feasible cells early on increases the confidence level of question 2. Level 1 gives a really nice, compact overview of the considered solutions. I can image that this can be an issue if no cells are feasible	Questions 3 and 4 can only be fully answered with the last level of transparency
B1	Grid size and max heading changes are shown only in the last level. That is good. I believe that an experienced ATC will naturally have a good feel for the maximum heading changes and the grid size, making it redundant to visualize them. The red and blue cells are (I think) and should be shown earlier as is done in Method B	
B2	The intent of the algorithm becomes apparent really quickly due to showing all the feasible cells	there are no feasible cells at a certain state band you wouldn't know some cells are evaluated without changing the transparancy.
B2	Because there is not a lot of information to process approximating what the algorithm is doing and why costs less effort	
B3	in some scenarios, the fact that there were less signs (dots, bands, squares) made it easier to see why some points were not used. Less information had to be processed to come to a conclusion. Thus I perceived it as an easier task.	lack of information at the early stages delayed the moment when I can confidently say what type of strategy was used/why a point was not selected
B4	Nice to see the feasible cells at lower levels	Takes more time to evaluate the solution space.
B5	Immediate presentation of feasible/selectable cells	Headings show up quite late (but are not that useful anyway)
B5	Details come up progressively in a logical order	

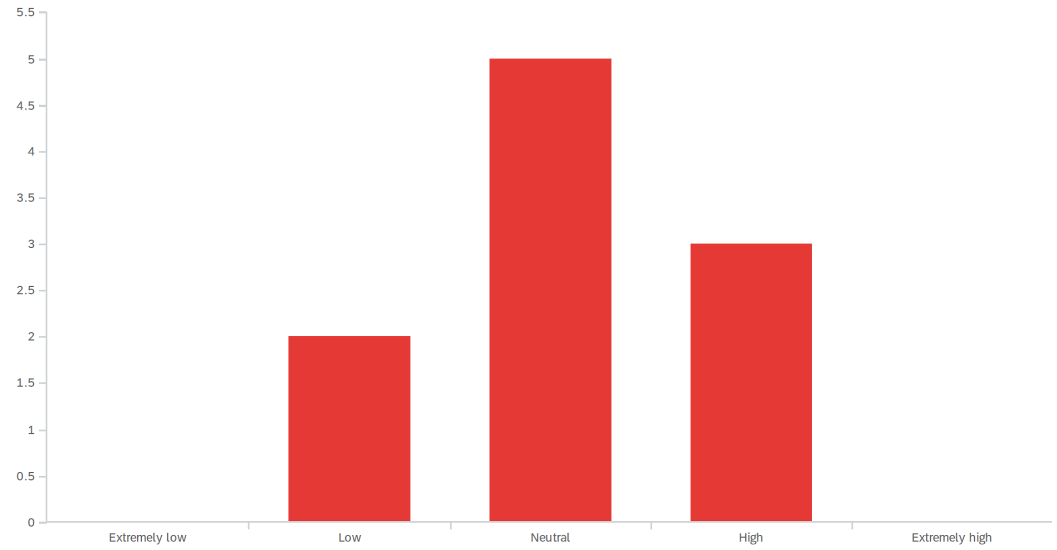


Figure F.1: Question: *How would you rate you workload during the experiment?* Y axis is number of responses.

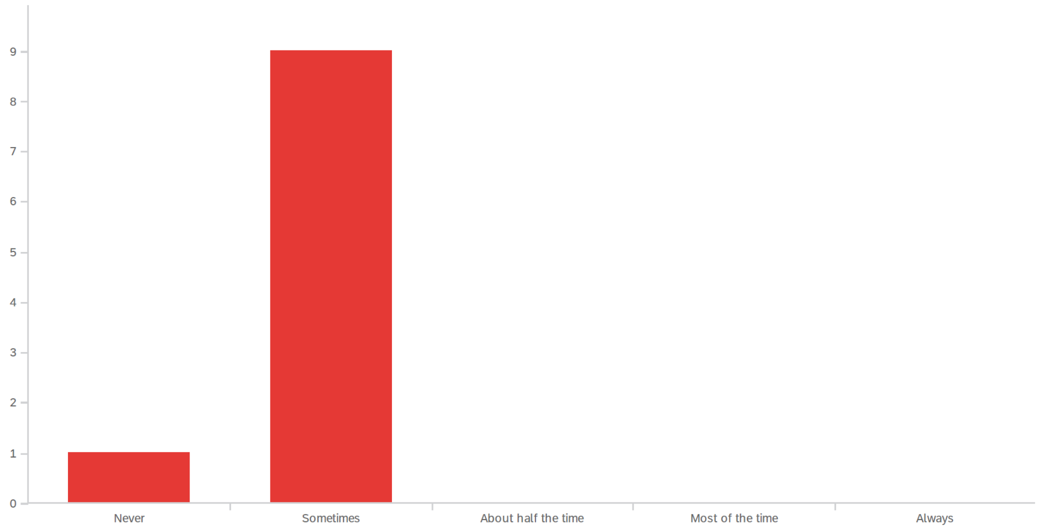


Figure F.2: Question: *Was information on the interface overwhelming?* Y axis is number of responses.

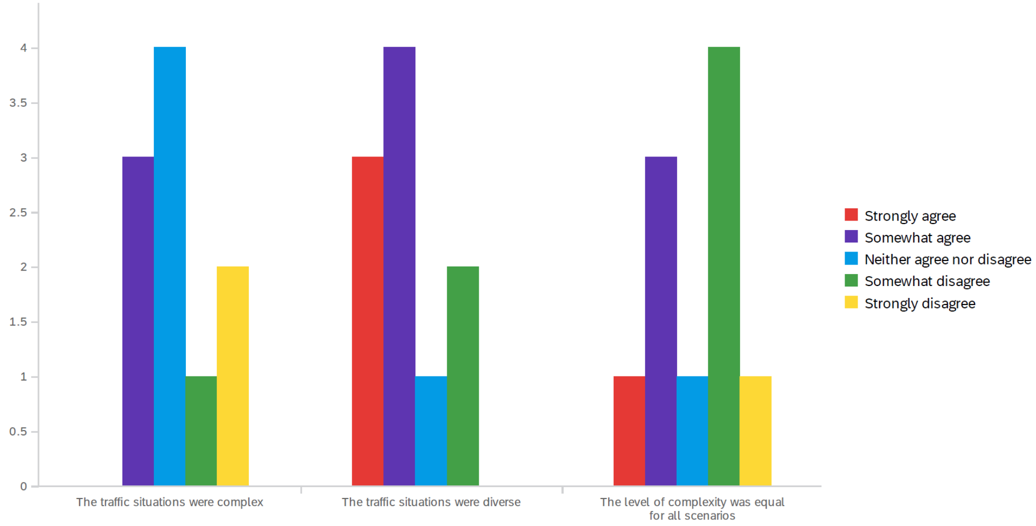


Figure F.3: Question: *Do you agree with the following statements?* Y axis is number of responses.

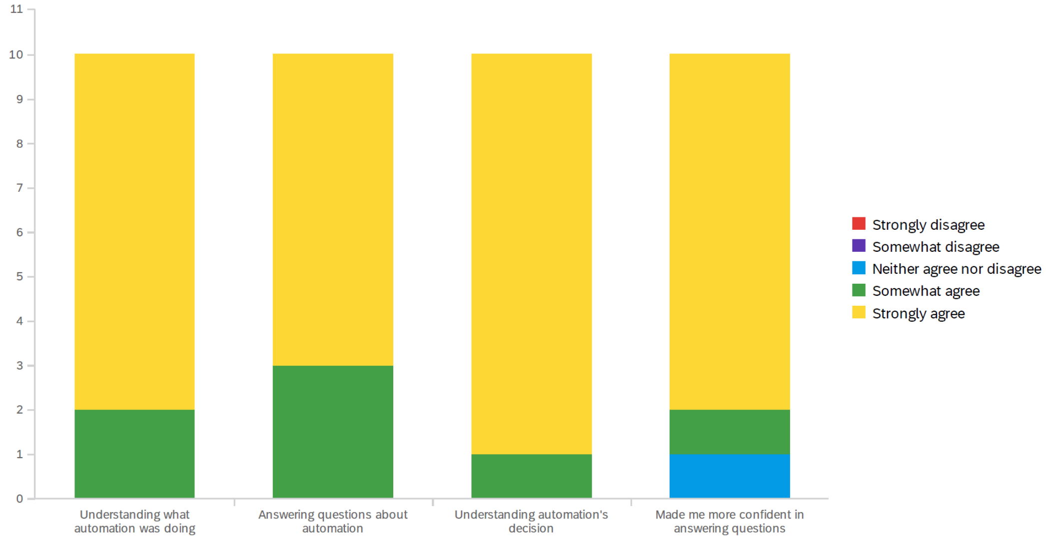


Figure F.4: Question: *Higher levels of transparency (compared to lower levels of transparency) helped me in .....?* Y axis is number of responses.

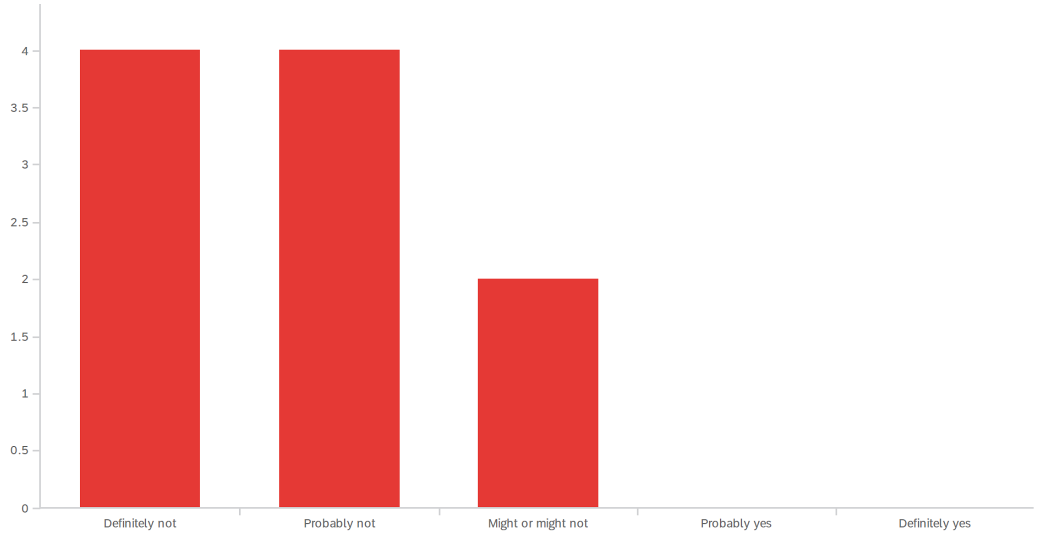


Figure F.5: Question: *At the highest level of transparency, was there any information missing?* Y axis is number of responses.

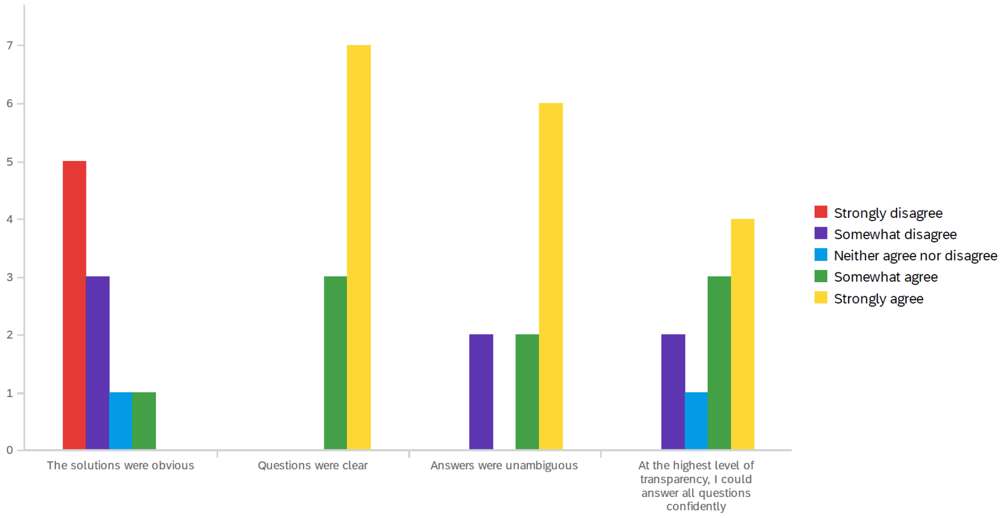


Figure F.6: Question: *Do you agree with the following statements?* Note that the questions of during the scenarios are meant here. Y axis is number of responses.

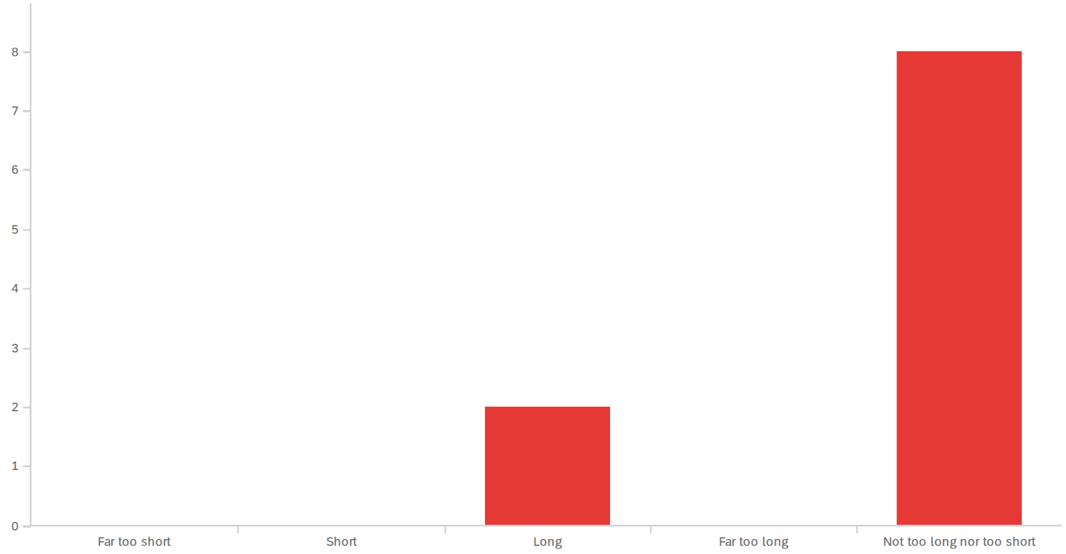


Figure F.7: Question: *What do you think of the length of the training?* Y axis is number of responses.

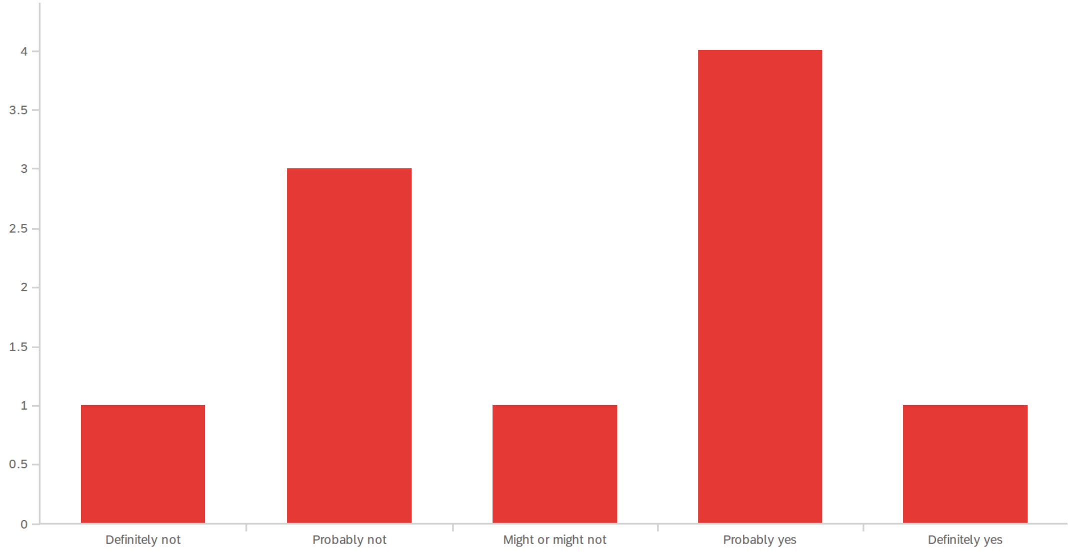


Figure F.8: Question: *Did you continue learning during the measurement phase of the experiment?* Y axis is number of responses.

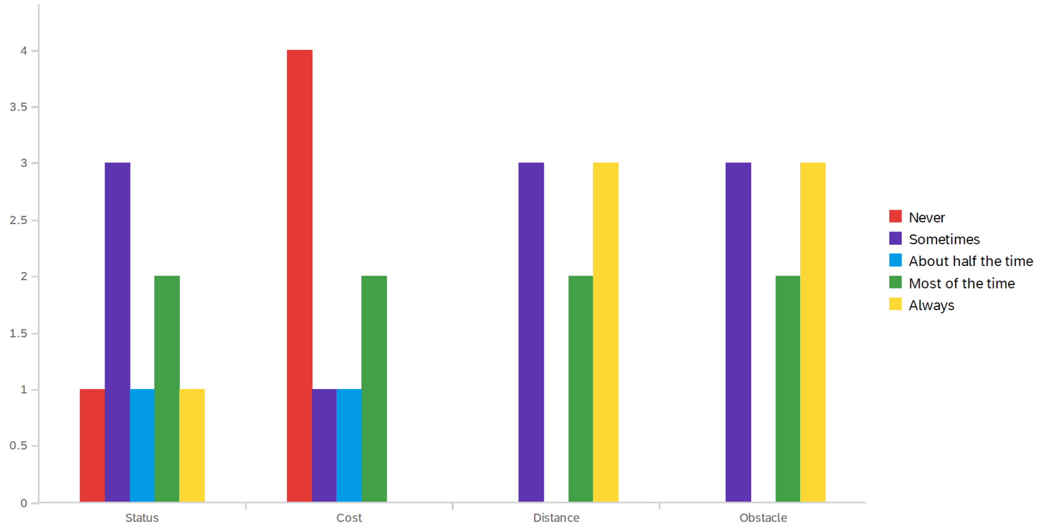


Figure F.9: Question: *What information of the label did you use?* Y axis is number of responses.

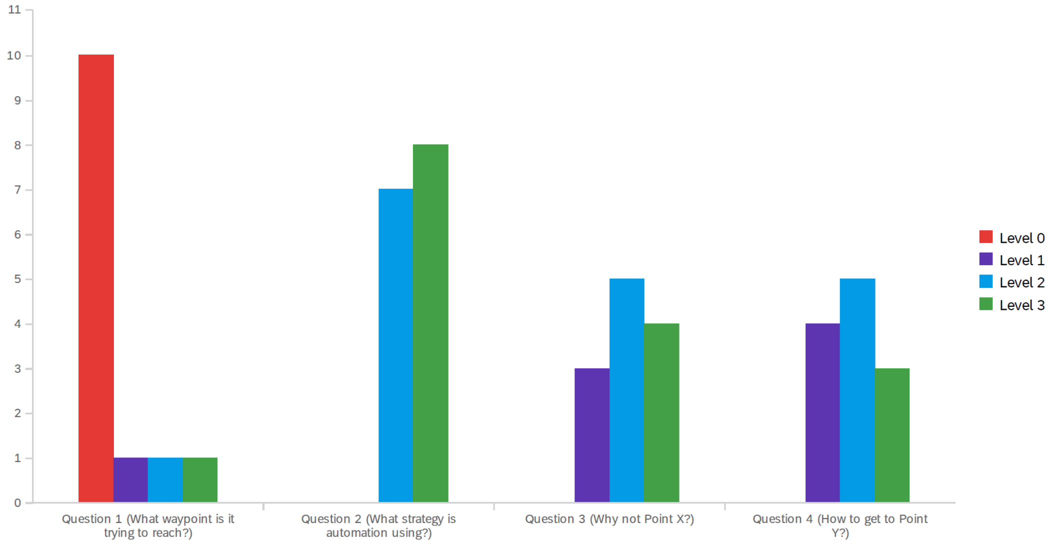


Figure F.10: Question: *What levels of the bottom-up approach did you use for answering questions?* Note multiple answers could be selected here. Y axis is number of responses.

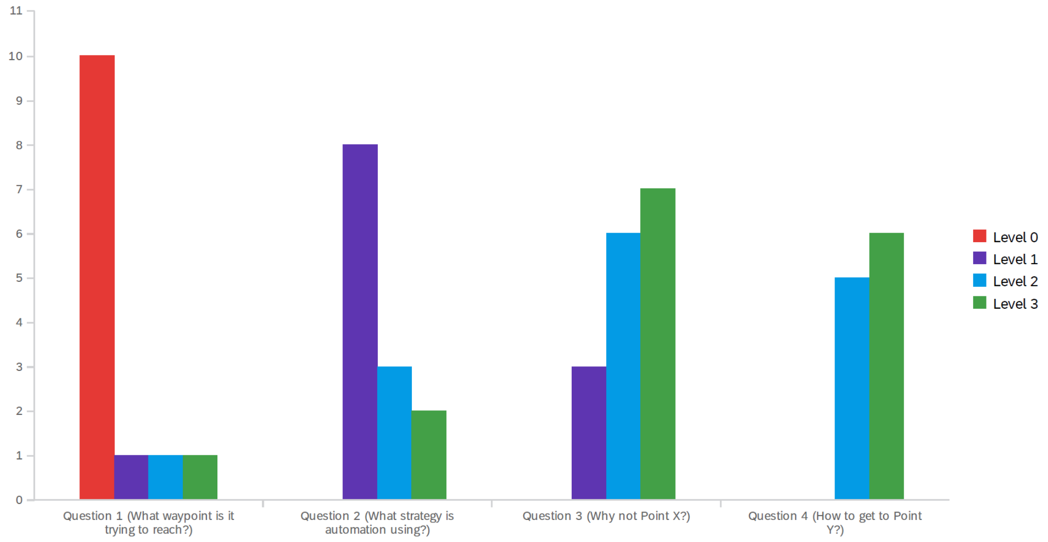


Figure F.11: Question: *What levels of the top-down approach did you use for answering questions?* Note multiple answers could be selected here. Y axis is number of responses.

Bibliography

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April, 2018. doi: 10.1145/3173574.3174156.

[2] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2870052.

[3] Ulf Ahlstrom. Work domain analysis for air traffic controller weather displays. *Journal of Safety Research*, 2005. ISSN 00224375. doi: 10.1016/j.jsr.2005.03.001.

[4] Mike Ananny and Kate Crawford. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media and Society*, 20(3):973–989, 2018. ISSN 14617315. doi: 10.1177/1461444816676645.

[5] L. Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983. doi: 10.1016/0005-1098(83)90046-8.

[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012.

[7] G. Baxter, J. Rooksby, Y. Wang, and A. Khajeh-Hosseini. The ironies of automation ... still going strong at 30? In *ACM International Conference Proceeding Series*, pages 65–71, 2012. doi: 10.1145/2448136.2448149.

[8] Adella Bhaskara, Michael Skinner, and Shayne Loft. Agent Transparency : A Review of Current Theory and Evidence. *IEEE Transactions on Human-Machine Systems*, 50(3):215–224, 2020. doi: 10.1109/THMS.2020.2965529.

[9] C. E. Billings. *Aviation automation : the search for a human-centered approach*. Human factors in transportation. Lawrence Erlbaum Associates Publishers, Mahwah, N.J., 1997. ISBN 0805821260 9780805821260 0805821279 9780805821277.

[10] Clark Borst. Shared Mental Models in Human-Machine Systems. *IFAC-PapersOnLine*, 49(19): 195–200, 2016. ISSN 24058963. doi: 10.1016/j.ifacol.2016.10.517.

[11] Clark Borst. AE4318: Supervisory Control Cognitive Systems - Cognitive Work Analysis, 2019.

[12] Clark Borst. Human-Automaion Interaction AE4316: Aerospace Human-Machine Systems, 2019.

[13] Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data and Society*, 3(1):1–12, 2016. ISSN 20539517. doi: 10.1177/2053951715622512.

[14] Jessie Y. C. Chen, Katelyn Procci, Michael Boyce, Julia Wright, Andre Garcia, and Michael J. Barnes. Situation Awareness–Based Agent Transparency. *US Army Research Laboratory*, 2014.

[15] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F. Maxwell Harper, and Haiyi Zhu. Explaining Decision-Making Algorithms through UI. *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019. doi: 10.1145/3290605.3300789.

[16] SESAR Consortium. *SESAR Definition Phase Deliverable 3 - The ATM Target Concept*. 2007. DLM-0612-001-02-00.

[17] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 2008. ISSN 09241868. doi: 10.1007/s11257-008-9051-3.

[18] Nicholas Diakopoulos. Accountability in algorithmic decision-making. *Queue*, 13(9), 2015. ISSN 15427749. doi: 10.1145/2857274.2886105.

[19] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959. ISSN 0029599X. doi: 10.1007/BF01386390.

[20] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. The role of trust in automation reliance. *International Journal of Human Computer Studies*, 2003. ISSN 10715819. doi: 10.1016/S1071-5819(03)00038-7.

[21] Malin Eiband, Hanna Schneider, Mark Bilandzic, Julian Fazekas-Con, Mareike Haug, and Heinrich Hussmann. Bringing transparency design into practice. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pages 211–223, 2018. doi: 10.1145/3172944.3172961.

[22] EUROCONTROL. Cross-border air navigation services, 05-10-2020 2020. URL https://www.eurocontrol.int/muac.

[23] John M. Flach, Kazuo Monta, Fumiya Tanabe, Kim J. Vicente, and Jens Rasmussen. Ecological approach to interface design. *Proceedings of the Human Factors and Ergonomics Society*, 1: 295–299, 1998. ISSN 10711813.

[24] Mahtab Ghazizadeh, John D. Lee, and Linda Ng Boyle. Extending the Technology Acceptance Model to assess automation. *Cognition, Technology and Work*, 2012. ISSN 14355558. doi: 10.1007/s10111-011-0194-3.

[25] R. Goritzlehner, C. Borst, J. Ellerbroek, Wes, M.M. van Paassen, and Max Mulder. Effect of ATC Automation Transparency on Acceptance of Resolution Advisories. *IEEE International Conference on Systems, Man, and Cybernetics*, 2014. URL https://linkinghub.elsevier.com/retrieve/pii/S2405896319319226.

[26] H.P Grice. Logic and Conversation, 1975.

[27] Sachin Grover, Chiara Pulice, Gerardo I. Simari, and V. S. Subrahmanian. BEEF: Balanced English Explanations of Forecasts. *IEEE Transactions on Computational Social Systems*, 6(2):350–364, 2019. ISSN 2329924X. doi: 10.1109/TCSS.2019.2902490.

[28] Riccardo Guidotti, Anna Monreale, and Salvatore Ruggieri. Factual and Counterfactual Explanations for Black Box Decision Making. *IEEE Computer Society*, pages 14–23, 2019.

[29] Benoit Guiost, Serge Debernard, Thierry Poulain, and Patrick Millot. Task allocation in air traffic control involving a common workspace and a cooperative support system. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2006. ISBN 9783902661050. doi: 10.3182/20060522-3-fr-2904.00015.

[30] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968. ISSN 21682887. doi: 10.1109/TSSC.1968.300136.

[31] Jacco Hoekststra and Joost Ellerbroek. AE4321 Air Traffic Management Human Machine Interface. Unpublished lecture slides, 2019.

[32] Jacco Hoekststra and Joost Ellerbroek. AE4321 Air Traffic Management Structure of Airspaces. Unpublished lecture slides, 2019.

[33] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for Explainable AI: Challenges and Prospects. 2018. URL `http://arxiv.org/abs/1812.04608`.

[34] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011. ISSN 01679236. doi: 10.1016/j.dss.2010.12.003.

[35] Husni R. Idris, Tarek El-Wakil, and David J. Wing. Trajectory planning by preserving flexibility: Metrics and analysis. *AIAA Guidance, Navigation and Control Conference and Exhibit*, (August): 1–14, 2008. doi: 10.2514/6.2008-7406.

[36] Husni R. Idris, Ni Shen, Tarek El-Wakil, and David J. Wing. Analysis of trajectory flexibility preservation impact on traffic complexity. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (August):1–14, 2009. doi: 10.2514/6.2009-6168.

[37] Husni R. Idris, Ni Shen, and David J. Wing. Improving separation assurance stability through trajectory flexibility preservation. *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, 1(September):1–11, 2010. doi: 10.2514/6.2010-9011.

[38] International Civil Aviation Organization. *Doc 4444 - Air Traffic Management - Procedures for Air Navigation Services*. 2016. ISBN 978-92-9258-081-0.

[39] Daniel P. Jenkins, Neville A. Stanton, Paul M. Salmon, and Guy H. Walker. *Cognitive work analysis: Coping with complexity*. Ashgate Publishing Limited, 2008. ISBN 9780754670261. doi: 10.1080/00140130903458293.

[40] Jiun-Yin Jian, Ann M. Bisantz, and Colin G. Drury. Foundations for an Empirically Determined Scale of Trust in Automated Systems. *International Journal of Cognitive Ergonomics*, 2000. ISSN 1088-6362. doi: 10.1207/s15327566ijce0401_04.

[41] Jakko Kemper and Daan Kolkman. Transparent to whom? No algorithmic accountability without a critical audience. *Information Communication and Society*, 22(14):2081–2096, 2019. ISSN 14684462. doi: 10.1080/1369118X.2018.1477967.

[42] Rene F. Kizilcec. How much information? Effects of transparency on trust in an algorithmic interface. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2390–2395, 2016. doi: 10.1145/2858036.2858402.

[43] Rolf Klomp, Clark Borst, Max Mulder, Gesa Praetorius, Martijn Mooij, and Dennis Nieuwenhuisen. Experimental evaluation of a joint cognitive system for 4D trajectory management. *SIDs 2013 - Proceedings of the SESAR Innovation Days*, (November), 2013.

[44] Rolf Klomp, Clark Borst, René Van Paassen, and Max Mulder. Expertise Level, Control Strategies, and Robustness in Future Air Traffic Control Decision Aiding. *IEEE Transactions on Human-Machine Systems*, 46(2):255–266, 2016. ISSN 21682291. doi: 10.1109/THMS.2015.2417535.

[45] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *Conference on Human Factors in Computing Systems - Proceedings*, 2012. ISBN 9781450310154. doi: 10.1145/2207676.2207678.

[46] Nathan Lau, Øystein Veland, Jordanna Kwok, Greg Jamieson, Catherine Burns, Alf Braseth, and Robin Welch. Ecological interface design in the nuclear domain: An application to the secondary subsystems of a boiling water reactor plant simulator. *IEEE Transactions on Nuclear Science*, 55, 12 2008. doi: 10.1109/TNS.2008.2005979.

[47] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2119–2128, 2009. doi: 10.1145/1518701.1519023.

[48] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, and Su-In Lee. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749–760, 2018. ISSN 2157-846X. doi: 10.1038/s41551-018-0304-0.

[49] Joseph B. Lyons, Garrett G. Sadler, Kolina Koltai, Henri Battiste, Nhut T. Ho, Lauren C. Hoffmann, David Smith, Walter Johnson, and Robert Shively. Shaping trust through transparent design: Theoretical and experimental guidelines. In *Advances in Intelligent Systems and Computing*, 2017. ISBN 9783319419589. doi: 10.1007/978-3-319-41959-6_11.

[50] G. A. Mercado Velasco, M. Mulder, and M. M. Van Paassen. Analysis of air traffic controller workload reduction based on the solution space for the merging task. *AIAA Guidance, Navigation, and Control Conference*, (August), 2010. doi: 10.2514/6.2010-7541.

[51] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 00043702. doi: 10.1016/j.artint.2018.07.007.

[52] Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. The ethics of algorithms: Mapping the debate. *Big Data and Society*, 3(2):1–21, 2016. ISSN 20539517. doi: 10.1177/2053951716679679.

[53] John Morris. Algorithm Animation: Using algorithm code to drive an animation. *Conferences in Research and Practice in Information Technology Series*, 42:15–20, 2005. ISSN 14451336.

[54] NASA. Human systems integration division @ nasa ames - home - downloads - tlx scale, Oct 2020. URL `https://humansystems.arc.nasa.gov/groups/tlx/downloads/TLXScale.pdf`.

[55] International Civil Aviation Organisation. *Annex 11: Air Traffic Services*. July 2018. ISBN 975-92-9258-481-8. URL `https://unitingaviation.com/publications/Annex-11/#page=55`.

[56] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 30(3):286–297, 2000. ISSN 10834427. doi: 10.1109/3468.844354.

[57] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. A slow algorithm improves users' assessments of the algorithm's accuracy. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), 2019. ISSN 25730142. doi: 10.1145/3359204.

[58] Pragya Paudyai and B. L.William Wong. Algorithmic opacity: Making algorithmic processes transparent through abstraction hierarchy. *Proceedings of the Human Factors and Ergonomics Society*, 1(1):192–196, 2018. ISSN 10711813. doi: 10.1177/1541931218621046.

[59] Jorge Pinto, Rolf E. Klomp, Clark Borst, Marinus M van Paassen, and Max Mulder. Design of an Ecological Flow-based Interface for 4D Trajectory Management in Air Traffic Control. *5th CEAS Air and Space Conference*, (141):1–10, 2015.

[60] Jens Rasmussen. *The human data processor as a system component. Bits and pieces of a model*. Number 1722 in Risø-M. Risø National Laboratory, 1974.

[61] Jens Rasmussen. Skills, Rules and Knowledge: Signals, Signs and Symbols , and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics*, 13 (3):257–266, 1983.

[62] Mitchel Resnick, Robbie Berg, and Michael Eisenberg. Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *Journal of the Learning Sciences*, 2000. ISSN 10508406. doi: 10.1207/s15327809jls0901_3.

[63] Thomas B Sheridan and William L. Verplank. Human and Computer Control of Undersea Teleoperators. Technical report, 1978.

[64] SKYbrary. Controller Pilot Data Link Communications, 06-10-2020 2020. URL `https://www.skybrary.aero/index.php/Controller_Pilot_Data_Link_Communications_(CPDLC)`.

[65] SKYbrary. Flight Progress Strips, 2020. URL `https://www.skybrary.aero/index.php/Flight_Progress_Strips`.

[66] SKYbrary. Tactical Controller Tool, 06-10-2020 2020. URL `https://www.skybrary.aero/index.php/Tactical_Controller_Tool_(TCT)`.

[67] D A Spencer. Applying artificial intellignece techniques to air traffic control automation. *The Lincoln Laboratory Journal*, 2(3):537–554, 1989.

[68] Jean Claude Sperandio. The regulation of working methods as a function of work-load among air traffic controllers. *Ergonomics*, 1978. ISSN 13665847. doi: 10.1080/00140137808931713.

[69] A Springer and S Whittaker. Making Transparency Clear: The Dual Importance of Explainability and Auditability. *Joint Proceedings of the ACM IUI 2019 Workshops*, 2019. URL `http://ceur-ws.org/Vol-2327/IUI19WS-IUIATEC-5.pdf`.

[70] A Springer and S Whittaker. Progressive Disclosure: Designing for Effective Transparency. *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019. doi: 10.4324/9781315101002-36.

[71] D. S.A. Ten Brink. Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control. Master's thesis, 2017. Unpublished.

[72] Oliver Turnbull and Arthur Richards. Examples of Supervisory Interaction with Route Optimizers. *Second SESAR Innovation Days*, (November), 2012.

[73] SESAR Joint Undertaking. *European ATM master plan*. Publications Office of the European Union, Luxembourg, 2020. ISBN 978-92-9216-134-7. doi: 10.2829/695700.

[74] Stijn B.J. Van Dam, Max Mulder, and M. M. van Paassen. Ecological interface design of a tactical airborne separation assistance tool. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 38(6):1221–1233, 2008. ISSN 10834427. doi: 10.1109/TSMCA.2008.2001069.

[75] M. M. Van Paassen, Clark Borst, Max Mulder, Rolf Klomp, Pim Van Leeuwen, and Martijn Mooij. Designing for shared cognition in air traffic management. *SIDs 2011 - Proceedings of the SESAR Innovation Days*, (January), 2011. ISSN 2211002X. doi: 10.3233/aop-130030.

[76] Ladislav Végh and Veronika Stoffová. Algorithm animations for teaching and learning the main ideas of basic sortings. *Informatics in Education*, 16(1):121–140, 2017. ISSN 16485831. doi: 10.15388/infedu.2017.07.

[77] Kim J Vicente. *Cognitive work analysis: Toward safe, productive, and healthy computer-based work.* Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US, 1999. ISBN 0-8058-2396-4 (Hardcover); 0-8058-2397-2 (Paperback).

[78] Kim J. Vicente and Jens Rasmussen. Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):589–606, 1992. ISSN 21682909. doi: 10.1109/21.156574.

[79] C. Westin, B. Hilburn, C. Borst, E. J. van Kampen, and M. Bång. Building transparent and personalized ai support in air traffic control. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–8, 2020. doi: 10.1109/DASC50938.2020.9256708.

[80] Carl Westin, Brian Hilburn, and Clark Borst. The effect of strategic conformance on acceptance of automated advice: Concluding the MUFASA project. *SIDs 2013 - Proceedings of the SESAR Innovation Days*, (January), 2013.

[81] Carl Westin, Clark Borst, and Brian Hilburn. Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation? *IEEE Transactions on Human-Machine Systems*, 46(1): 41–52, 2016. ISSN 21682291. doi: 10.1109/THMS.2015.2482480.

[82] Carl Westin, Clark Borst, and Brian Hilburn. Automation Transparency and Personalized Decision Support: Air Traffic Controller Interaction with a Resolution Advisory System. *IFAC-PapersOnLine*, 49(19):201–206, 2016. ISSN 24058963. doi: 10.1016/j.ifacol.2016.10.520.

[83] Christopher D Wickens. *An introduction to human factors engineering*. Pearson Prentice Hall, 2nd ed. edition, 2004. ISBN 0131229176 9780131229174.

[84] Jenna Wiens and Erica S. Shenoy. Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology. *Clinical Infectious Diseases*, 2018. ISSN 15376591. doi: 10.1093/cid/cix731.