A COMPUTATIONAL APPROACH TO SITUATION AWARENESS AND MENTAL MODELS IN AVIATION

A Thesis
Presented to
The Academic Faculty

by

Sebastien Mamessier

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2013

A COMPUTATIONAL APPROACH TO SITUATION AWARENESS AND MENTAL MODELS IN AVIATION

Approved by:

Dr Karen Feigh, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr Amy Pritchett
School of Aerospace Engineering
Georgia Institute of Technology

Dr Ashok Goel
School of Interactive Computing
Georgia Institute of Technology

Date Approved: June 28, 2013

ACKNOWLEDGEMENTS

It is with immense gratitude that I acknowledge the support and help of my committee chair and advisor Dr Karen Feigh. She spent an enormous amount of time guiding me through the maze of my thoughts and the writing of this thesis, learning french syntax along the way. She always knew how to keep me motivated and help me raise the quality of my work. Thanks to her, I learnt a lot about human-automation as well as human-human interaction and I'm really thankful for everything. I'm also grateful to my committee member Dr Amy Pritchett who was always available to share her expertise and provided me with priceless advice along this work. I would also like to thank my committee member Dr Ashok Goel for sharing with me his time and wisdom. Our long conversations were always worth the trip and the parking tickets. I hope we will happen to work together again.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

Although most modern, highly-computerized flight decks are known to be robust to small disturbances and failures, humans still play a crucial role in advanced decision making in off-nominal situations, and accidents still occur because of poor human-automation interaction. In complex safety critical human integrated systems, loss of control or automation surprises often trace their origin to disparities between the human agent's mental representation of a system and the actual states/dynamics of the system and its environment [4, 21]. In addition to the physical state of the environment, operators now have to extend their awareness to the state of the automation itself which supposes an active monitoring action [24, 25]. To guarantee the accuracy of this knowledge, humans need to know the dynamics or approximate versions of the dynamics that rule the automation. A general insight about Newton's laws and good monitoring skills are no longer sufficient to guarantee completely safe behaviors of such systems involving complex automation-human interactions [24, 25]. Comprehensive and detailed training about automated flight systems is also not sufficient since an excessive complexity might still lead the pilot to misunderstand the automation's behavior.

The operator's situation awareness can decline because of a deficient mental model of the aircraft and an excessive workload. A local absence of knowledge or some confusion due to too much complexity can be responsible for a faulty mental model.

Excessive workload can be caused by the operator's attempt to understand the automation's complex behavior in off-nominal situations or by a poorly designed partition of tasks. Generally, automated systems and related training programs should be analyzed in the light of their impact on the operator's mental model and situation awareness.

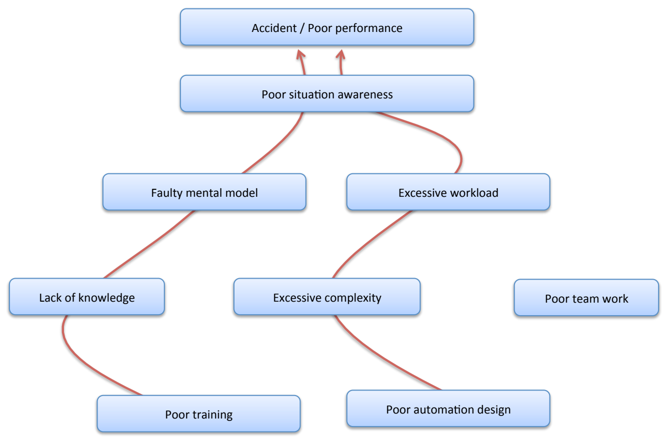

Figure 1: Non-environmental causes and effects of accidents due to poor human-automation interaction. The paths are a possible summary of AF447's final report conclusions.

Usual mechanisms to investigate the influence of new automation on human automation interaction includes extensive use of simulation and human-in-the-loop (HITL) experimentation. But HITL experiments are expensive, time-consuming and consequently can only evaluate a limited set of scenarios. Therefore we need to improve our simulation capabilities by developing realistic, computational human agent models that account for human limitations.

This work describes the creation of a computational human agent model simulating the cognitive constructs of *situation awareness* and *mental models* known to capture the symptoms of poor human-automation interaction and provide insight into more comprehensive metrics supporting the validation of automated systems in aviation.

1.2 Requirements

Concepts of mental models and situation awareness are broad and ill-defined in the literature.To address a computational approach, this work will need to clearly define and scope the terminology. As situation awareness and mental models will be modeled and implemented to support the simulation of HAI as comprehensively as possible, the operational definition of these concepts will project theoretical notions onto a set of computational objects. The implementation should allow the analysis of aviation incidents such as the crash of the flight AF 447 which gives important examples of poor human-automation interaction, excessive workload, lacunary mental models and resulting low situation awareness. Such a case study can therefore be used to provide a list of requirements, scope concepts and define objectives about what needs to be included in the computational agent model. The accident report of the Bureau dEtudes des Accidents (BEA) mentions the unusually high workload throughout the document. This high workload is said to be partially responsible for the degradation of the situation awareness and communications between pilots. A first requirement would be to have an accurate and reliable indicator for workload. Then, should it be degraded by a high workload or a loss of sensors, the report points out the importance of the pilot's mental representation of the aircraft's state. Therefore, our system should include constructs allowing analysts to diagnose situations where the pilot's belief is clearly different from the actual aircraft situation. This capacity should be encapsulated within the broader concept of Situation Awareness as a mirrored mental version of the aircraft state, sampled from the instruments through observation. In the absence of instruments, pilots rely on their knowledge of the dynamics of the plane to analyze the situation. The report stated a lacunary knowledge of high atmospherical flight dynamics by one of the crew members that could explain a poor situation awareness. Thus the agent model has to take into account the pilot's mental model of continuous aircraft dynamics to predict and prevent this kind of accident. Another possibility to support the accident scenario is the absence of acknowledgement from the pilots of the system switching to an alternate control law as a consequence of the loss of airspeed indication. High workload or lacunary knowledge of mode transitions might be responsible for such a misunderstanding. This latter interpretation adds discrete dynamics such as autopilot modes and their knowledge by pilots as an important part of mental models influenced by the current workload with a strong impact on situation awareness. Therefore, *mental models* of discrete dynamics must also be accounted for. Finally, the transcript of the communications between pilots indicates that the pilots lost confidence in their instrument and in their ability to identify the problem. The computational agent model described in this work should discuss and model the relative confidence maintained by pilots of their own belief of the aircraft state against their observations as well as a method to optimally fusion both sources of information.

1.3 Background and Definitions

A computational approach of roughly bounded cognitive concepts often finds itself striving to achieve two conflicting goals: comprehensiveness and operationalizability. Therefore, the first objective of this work is to conceptually define and clearly bound the underlying concepts and their interactions. How does *situation awareness* interact with the *mental model*. To what extent are they conceptually different? How does learning shape mental models? Can clear non-overlapping definitions be derived? The second objective is to identify, for each concept, key components and their relationships. Which part of *situation awareness* is impacted by *mental models*? How does SA provide a feedback to alter mental models? And even within SA, which level relies on another level? After mental models and learning processes have been clearly defined, their key components and interactions identified, the third objective is to find reasonable compromises between preserving the original theoretical concepts and satisfying the implementation constraints. Since this work is part of a larger project, the result should also fit into the general work modeling philosophy.

1.3.1 Background - Situation Awareness

In 1988, Starter and Woods introduced the concept of situation awareness without the support of an accurate definition [33]. In 1995, situation awareness was extensively studied by Endsley [9, 8] and given a formal definition, broken down into three different levels describing perception (L1), comprehension (L2) and anticipation (L3). The literature contains many examples of methodologies supporting the measurement of situation awareness *in situ* including the "situation awareness global assessment technique" (SAGAT) developed by Endsley [7, 9]. Such measurement methods provide insight but do not directly indicate how to model situation awareness in computational simulations. That situation awareness is believed to be measurable by freezing a task [7] and simply querying the operator's knowledge is encouraging. Indeed, it means that situation awareness is intelligible and might be subject to simulation. However, a computational framework for simulating situation awareness is missing and an attempt to sketch the outlines of such an approach will be a part of this work. The definition we will endorse in that work is a synthesis of the literature review adapted to our computational needs :

Definition 1.1 *The operator's situation awareness describes his or her subjective understanding of the current state of the world which encompasses the immediate belief of the state and the degree of confidence about this state.*

1.3.2 Background - Mental models

Starter et al. mention that the distinction between *situation awareness* and the concept of *mental models* seems to be blurred by the work of different authors [33]. Both concept do not have the same point of reference and "adequate mental models are one of the prerequisites for achieving situation awareness." Indeed *mental models* support the acquisition and maintenance of *situation awareness*. Endsley pointed out that: “It is first necessary to distinguish the term situation awareness, as a state of knowledge, from the processes used to achieve that state”. This work fully embraces this remark; this chapter focuses on a computational approach to model a human agent’s knowledge and understanding of the state of the world whereas processes used to maintain and achieve situation awareness will be developed further in the following chapters under the term *mental models*. This section reviews the literature about mental models to try to clarify this distinction.

In 1961, Forrester defined *mental models* as a “Verbal description, mental image” of the system [11]. Ten years later he added “Selected concepts and relationships used to represent the real system.” This image of *mental models* as a mental representation of the system is generally agreed upon [11, 26, 23, 31, 6], but the definitions differ on the boundaries. Richardson [31], for example describes a mental model as a set of different types of cognitive structures including perception and decision making whereas Morecroft limits his concept to one particular type [23]. These discrepancies between definitions led Doyle to qualify *mental models* as “ill-defined” by the system dynamics community [6]. Another important contributor to the *mental model* literature is Johnson-Laird whose work treats the construction and manipulation of mental models in a context of probabilistic thinking. Other cognitive psychologists discuss the use of such mental models [6, 14] especially regarding expert human agents. On the other hand, Human Computer Interaction(HCI) is a field that reached a rough consensus regarding *mental models*. HCI practitioners were able to explain errors that novices make while interacting with automation by using incorrect *mental models* [27]. Javaux explained Sarter and Wood’s results regarding human-automation interaction in highly-computerized aircraft [34, 35] introducing metrics to measure the complexity of mental models [17].

Synthesizing this literature review and adapting it to our computational needs leads to defining mental models as follows :

Definition 1.2 *The operator's mental model of a system is a subjective version of the system's discrete and continuous dynamics representing his/her current knowledge of the system.*

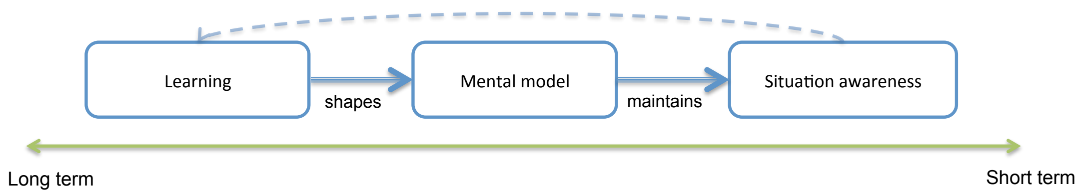

Figure 2: Preview of the overall structure of this work

1.4 Potential contributions

1.4.1 A human integrated work modeling framework

The cognitive engineering community uses *work* as a unit of analysis of socio-technical systems involving Human-Automation Interaction where *work* is defined as a purposeful activity acting on a dynamic environment and in response to the demands of this environment [28, 30, 1, 37]. Cognitive Work Analysis (CWA) is a comprehensive work modeling framework providing several methods to analyse the work domain, tasks, team cooperation and other cognitive aspects. However, such a framework does not provide computational means to verify the consistency of the system or the relevance of the team design. Pritchett et al. [28] formulated a list of functional requirements of a modeling framework to support the design of socio-technical systems such as the suitability to computational simulation to assess emergent behaviors and the ability to capture the way agents abstract the work. These arguments led to the creation of Work Models that Compute (WMC) [28]. This work will build on this modeling framework, extend it to implement the concepts of *situation awareness* and *mental models* and provide an easy way to design and analyze human integrated systems, output metrics related to performance and human agents, and identify potential poor design as far as human-computer interaction is concerned.

1.4.2 Evaluating training design and system complexity

Most accident reports from agencies such as National Transportation Safety Board (NTSB) or the french Bureau d'Etudes des Accidents (BEA) regarding accident investigations involving human-automation interaction incidents contain recommendations to improve the training procedures such as adding scenarios to simulator sessions, reinforcing the knowledge of specific procedures and automations behavior. However, the more training is needed to manage the system's automation, the more workload the operator undergoes to adequately monitor and understand the current state. The pilot might understand and know perfectly the dynamics ruling the flight deck system but his limited mental resources might not be sufficient to retrieve and process this knowledge, especially in off-nominal situations. Therefore, adding more training is not always a good solution. By modeling and capping the workload and including training as a design variable, we can identify flaws that prevent the validation of systems requiring unreasonable training and potentially generating a dangerous amount of workload. This work shows that the simulation of *mental models* and *situation awareness* provide more realistic means to generate and measure workload as well as the consequences of high workloads.

1.4.3 Evaluating operational procedures accounting for implicit learning

Although an agent's knowledge of a system highly depends of the initial training phase, agents' representation of complex systems also changes over time as a result of daily procedures and experience. This phenomenon is also called *implicit learning* or at the neurologic level *hebbian learning*, a theory that explains how repeated scenarios strengthen or weaken the knowledge of certain rules and therefore modify the operator's mental model on a long term fashion. Implicit learning is not well captured by HITL experiments since it requires simulating thousands of hours of operation. Javaux [16] describes a computational method to simulate the impact of *implicit learning* on pilots' knowledge of autopilot modes and transitions. This work proposes to integrate and implement Javaux's approach into a broader design and analysis framework to study potential interactions between automation design, operational procedures and identify the emergence of dangerous situations and implicit learning patterns.

1.4.4 Evaluate new control algorithms versus pilot's situation awareness

Pilots have a good knowledge of flight dynamics and a reasonable understanding of linear control principles that allow them to fly manually and operate the autopilot safely. Incidents can still occur in off-nominal situations where the autopilot is automatically shut down or is controlling a damaged aircraft. Researchers are working on loss-of-recovery control algorithms that would allow to automatically or semi-automatically recover from loss-of-control situations. These algorithms are highly non-linear since they operate in emergency situation and their introduction requires new validation metrics. One of these new metrics should describe the safety of the interaction of such a system with the pilot to prevent misunderstanding and poor situation awareness. This work develops a method to evaluate the *situation awareness* of pilots as a reaction to these new algorithms in off-nominal situations where measuring workload and accounting for human factors is essential and which could lead to more comprehensive validation metrics.

1.5 Overview

As a summary, the goal of this work is to define a computational method to simulate *situation awareness* and *mental models* and integrate them into the overarching approach used by WMC to design and analyse socio-technical systems. Chapter II describes the principles of work simulation and details the logic behind the modeling and simulation framework WMC and Chapter III presents a test case. Chapter IV introduces the concept of *situation awareness* as a frame to model the agent's understanding, and proposes a computational approach. Chapter V focuses on the operators mental model as far as discrete dynamics are concerned whereas Chapter VI covers continuous *mental models*. Finally, Chapter VII summarizes how this work impacted the development of WMC.

CHAPTER II

SCOPE AND EXISTING FRAMEWORK

This thesis is part of a broader research project whose purpose is to simulate work in complex socio-technical systems through the development of the modeling and simulation framework Work Models that Compute (WMC). Following cognitive engineering design concepts, this framework separates the modeling of the work to be done from agent models. The work is modeled as a collection of actions and the environment by resources. At runtime, automation and human agents receive actions from *work models* and execute them according to their workload limitations.

The latter modeling structure has many advantages but fails to capture behaviors based on a subjective representation of the world. To account for agents with a different training or experience, part of the knowledge of the work has to be contained in the human agent models and must be vulnerable to potential long-term modifications. This subjective representation of the work constitutes the agents *mental model*.

Accounting for subjective *mental models* allows the development of new validation methods for highly automated systems, particularly in loss-of-control scenarios. In such off-nominal situations, the understanding of the pilot is crucial and has to be accounted for in simulation. This suggests the implementation of additional mental constructs within agent models sampling the actual state the world and providing a subjective base for decision making.

This work proposes a rigorous, generic method to allow the work to be subjectively executed by human agents accounting for different training and operational experience. The knowledge of the work reaches the human agent at two levels: the long-term knowledge of the dynamics of the system that can be lacunary, i.e mental models and the short-term understanding of the situation comprised of subjective copies of the current state of the world also called *situation awareness*.

2.1 Modeling situated work - Work Model that Computes

The cognitive engineering community uses *work* as a unit of analysis of socio-technical systems involving Human-Automation Interaction where *work* is defined as a purposeful activity acting on a dynamic environment, and in response to the demands of this environment [28, 30, 1, 37]. This definition is a starting point for Rasmussen and later Vicente who developed an extensive theory of cognitive system engineering called Cognitive Work Analysis (CWA) that provides several modeling methods allowing different analysis such as abstract multi-level modeling, task analysis and team design. However, such a framework does not provide quantitative means to verify the consistency of the system or the relevance of the team design. This type of analysis does not support the identification of unexpected behavior emerging from low-level constraints such as workload saturation. Pritchett et al. [28] formulated a list of functional requirements of a modeling framework to support the design of socio-technical systems such as the suitability to computational simulation to assess emergent behaviors and the ability to capture the way agents abstract the work. These arguments led to the creation of Work Models that Compute (WMC) [28]. Feigh et al. also used WMC and focused on the development of an advanced human agent model [10]. Although these papers present a quantitative method that allows the investigation of poor human-automation interaction and identify some emergent behaviors through simulation, the agent model currently included in WMC neither maintains a mental representation of the world nor implements a subjective understanding of the system and does not comprehensively implement concepts such as perception, comprehension of the situation, and projection of the agents in a near future. This chapter will describe several concepts necessary to fully understand this thesis and the constraints imposed by such a modeling approach that had to be taken into consideration.

2.2 Work models and agent models in WMC

In WMC, the product of the modeling of a socio-technical system is called a *work model*. It is independent of any agent's implementation and is to be created by the designer of the system. A work model includes several different levels of modeling, as shown in figure 3 and eventually results into the creation of *resources* that describe the state of the system and its environment and *actions* which are the low-level representation of the work and act on resources.

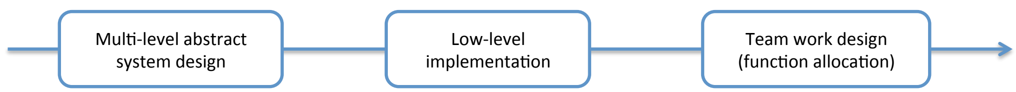

Figure 3: The creation of a work model can be broken down into three steps.

On the other hand, WMC allows the modeling of agents' behavior independently of any socio-technical system. The result of this effort is called an *agent model* and can be plugged in to any work model for simulation purposes. The agent model does not contain any information about the work but describes how the agent executes the actions whose actual content, i.e which *resources* are modified is received from the *work model* during the simulation. Figure 4 provides a comprehensive overview of the interaction between WMC's modeling components.

2.2.1 Work modeling

2.2.1.1 Work domain description

The first step to design a work model is to create a multi-level abstraction hierarchy introduced by Rassmussen [29] that represents the work from expressing the goals at

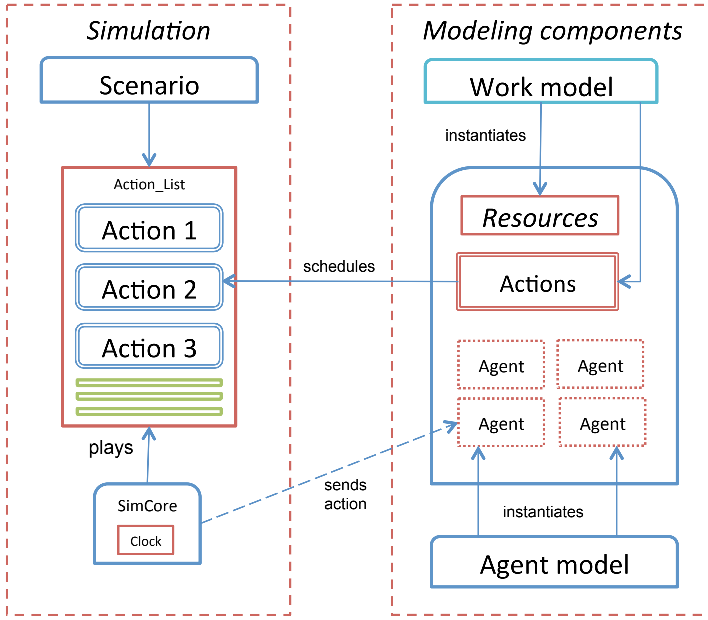

Figure 4: This diagram illustrated WMC's simulation and modeling components and how they interact.

a high-level point of view to the formulation of intermediary functions and a low-level enumeration of actions as shown in Figure 5. This representation helps the designer to identify the interactions between different levels of modeling and serves both task analysis and team design.

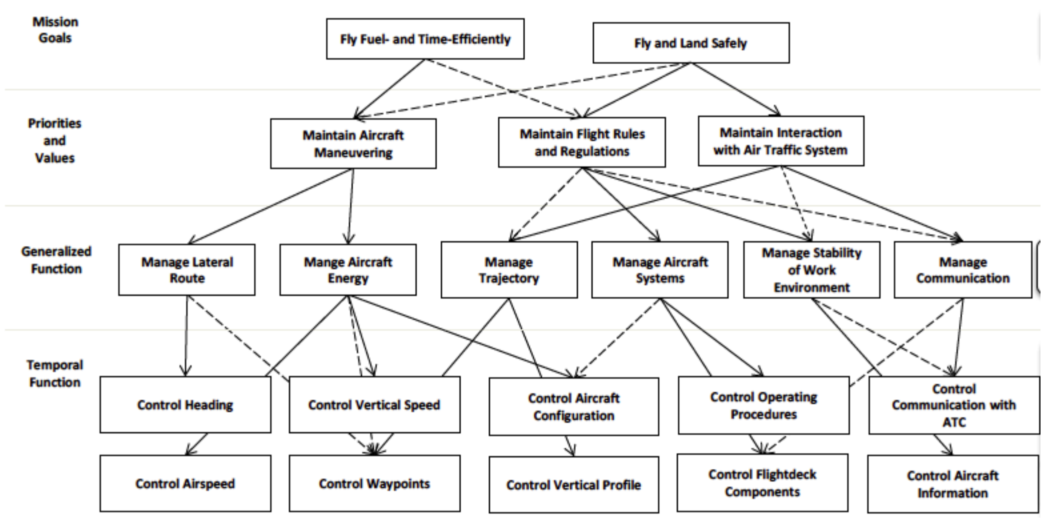

Figure 5: Creating the abstraction hierarchy is the first step in the creation of a work model. This figure shows the abstraction hierarchy of an aircraft+crew system

2.2.1.2 Low-level environment description: actions and resources

The lowest level of the abstraction hierarchy is used to create the resources needed to describe the environment and the tasks/dynamics that need to be executed to simulate the system and achieve the overarching goals.

Resources can be computationally represented by any type of variable (double, integer, boolean) and can describe either the state of the system like the `airspeed` of an aircraft or higher level information such as the `currentConfiguration` of a nuclear plant.

Actions are responsible for reacting to changes in the environment, i.e *resources* and changing them. For instance, *Actions* representing system dynamics will take a subpart of the world's state as input and update it according to a set of differential equations. On the other hand, *actions* acting on configuration variables will analyze the values of several resources and potentially change the system's setup such as the teamwork strategy.

2.2.1.3 Team work, function allocation

Finally, the *work model* needs to assign all the actions to the agents available. System dynamics of vehicles will be assigned to a simple agent whereas actions that implement some piece of analysis can be either given to human operators or automation agents. This step of the work modeling is called *function allocation.* Kim's work [18] is a good example of how to study function allocation with WMC. Function allocation can also be changed during the simulation responding to changes in the general strategy.

2.2.2 Agent modeling

As illustrated by figure 4, agent models are separated from the work modeling. Therefore the agent modeling task is narrowed down to the description of how the agent manages the actions that has been assigned to it. An automation agent might be modeled as executing incoming actions almost immediately where a human agent is likely to prioritize and delay actions according to the current workload. As a summary, agent models do not describe the content of the actions but the constraints and mechanisms responsible for actual execution. Section 2.3 will describe the human agent model in detail.

2.3 Human agent modeling

Several human agent models are available in WMC from a very basic human agent to an advanced performance model. Human agent models describe how an action received from the work model is handled. The action contains the description of the task itself and handles to the input and output resources. Basic human agents in WMC are comprised of several constructs illustrated in Figure 6 : an active action list, a delayed and an interrupted action list. Actions can be transferred from one list to another according to their priority attribute and an eventual saturation of each list. More advanced human agents can also forget actions that have been delayed or interrupted for too long as well as identifying upcoming actions and eventually change of strategy.

This advanced agent model allows the measurement of metrics such as the total workload as a function of time or the workload per type of action (monitoring, actual taskload, teamwork).

A big part of this work is an effort to make the human agent models more realistic and diagnostic by giving them the ability to learn about the *work model* that they are interacting with by allowing the simulation of situation awareness and evolution of mental models.

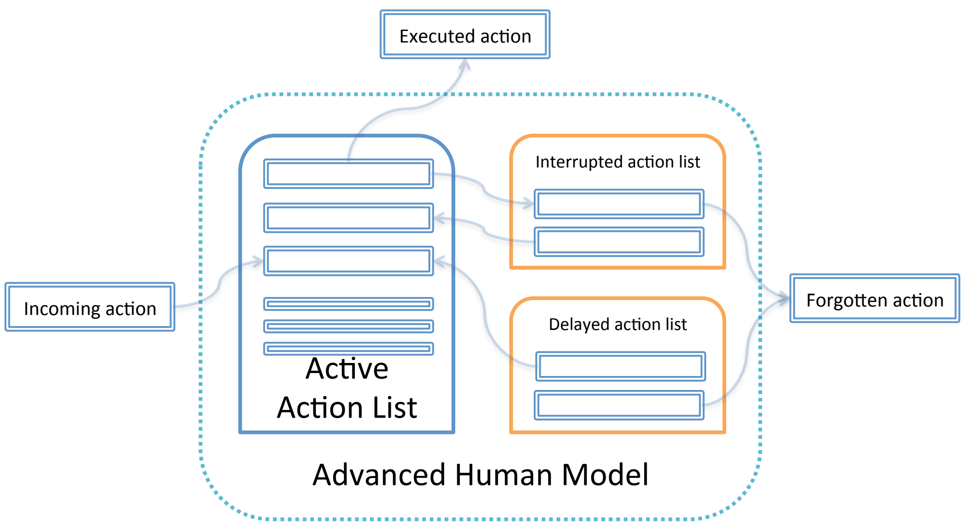

Figure 6: The creation of a work model can be broken down into three steps.

CHAPTER III

TEST CASE

3.1 *Case study : Boeing 747-400 in continuous descent approach at LAX airport*

The main field of application of this work is human-automation interaction with a particular interest in HAI in the flight deck. Therefore a *work model* describing an aircraft and its flight crew was chosen to evaluate this work. The *work model* presented in this section has been adapted from the Boeing 747-400 + crew *work model* used in [28, 18, 10]. Several modifications that were made to serve the purpose of this work will be described as well.

3.1.1 Description of the work model

The abstraction hierarchy of the continuous descent arrival aircraft model is comprised of two main goals: `Fly Fuel and Time Efficiently` and `Fly and Land Safely`. Both are then decomposed into lower level goals such as `Maintain Aircraft Maneuvering` which is further broken down into two functions: `Manage Lateral Route` and `Manage Aircraft Energy`. The lowest level of the hierarchy contains the actions for which some examples are given in the abstraction hierarchy shown in Figure 7.

The pitch and auto throttle autopilot modes of the Boeing 747-400 have been implemented as well as the different engagement/disengagement actions available on the Mode Control Panel (Figure 8). Automated mode activation in the Approach mode is also part of the work model.

A guidance module is present and allows modes such as VNAV to fly a lateral and vertical descent profile from a sequence of waypoints pre-computed by the Flight

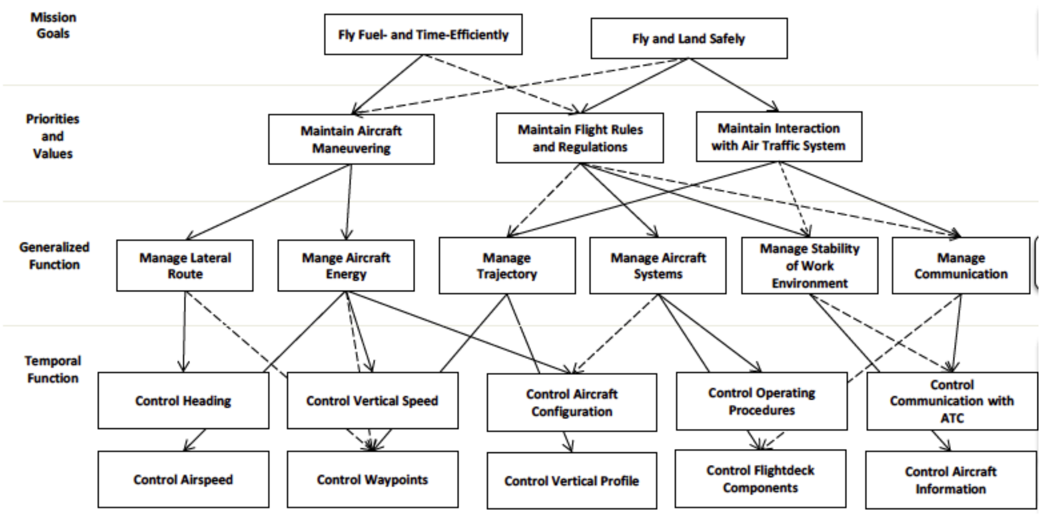

Figure 7: Portion of the abstraction hierarchy of the aircraft model used.

Management System.

The actions and functions handling Air Traffic Control(ATC) operations used to be implemented within the same model to study function allocation in Kim's work [18]. Since then, the ATC module was separated and now exists as a stand-alone *work model* that communicates with whatever aircraft located in a given sector. This effort along with the creation of *communication actions* made the interaction between Aircraft and ATC more realistic and supports large-scale simulation with multiple controllers and aircraft.

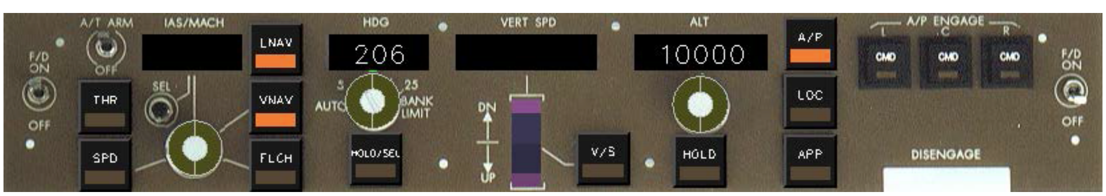

Figure 8: Mode Control Panel (MCP) used by pilots to engage/disengage pitch and autothrottle autopilot modes.

3.1.2 Scenario

The scenario on which the Aircraft *work model* is being tested is a continuous descent approach (CDA), as part of the Next Generation Air Traffic Operations principles. The aircraft and its crew operate on the RIIVR2 Standard Terminal Approach Route (STAR) of Los Angeles International Airport (LAX).

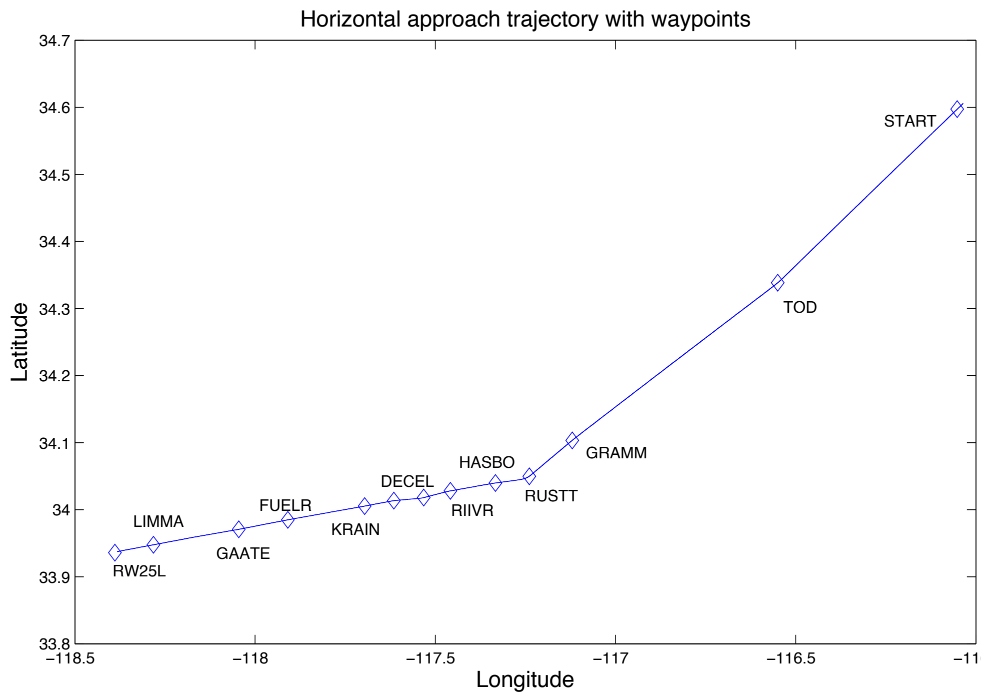

Figure 9: Trajectory obtained with WMC, START is the first waypoint, TOD is the top of descent and RW25L represents the runway.

Kim [18] tested four different task partitions between the pilot and the automation as part as her function allocation analysis, this thesis uses the most automated allocation as the basis for evaluation the impact of this work since the growing importance of automation is believed to induce a poor situation awareness in off-nominal situations. In the nominal descent scenario, controllers will clear the aircraft to the next altitude a few miles before reaching the next path leg. If a late clearance is issued, the aircraft will level off at the lowest altitude it has been cleared to until the reception of the clearance, once clear the aircraft will catch up on the descent path with a higher vertical speed and airspeed.

3.1.3 Mental model in a monitoring role

During this simulation, the pilots do not touch the yoke and only monitor the behavior of the automation in the nominal scenario. The crew might have to dial and press buttons on the MCP in response to unexpected ATC requests, but that is the extent of the pilot's interaction with the cockpit. Since no manual piloting is involved, this work does not include manual control schemes as part of the mental model of the pilot. Moreover, we make the reasonable assumption that pilots do not have a precise knowledge of the optimal control dynamics of the autopilot but maintain a good comprehension of which control loops are active in a specific mode and their approximate impact on the aircraft's motion. For example, the fact that both autopilot pitch and throttle inputs are used in `VNAV_SPD`, whereas no speed protection is offered by the `V/S` pitch mode are assumed to be known and part of the initial mental model of the flight crew since they appear in the operational training manuals.

While operating with the autopilot, the crew monitors variables of interest and compares them against expectations issued from the knowledge of the current autopilot mode. In the vertical navigation modes, the autopilot uses a combination of thrust and pitch inputs to reach or maintain a target state value (level,climbing,descending) with optimal performance while respecting the flight envelop. When the Vertical Navigation auto flight system is on, the crew is less likely to consider the open-loop dynamics of the aircraft as this would require an active monitoring of the values of inputs calculated by the autopilot which are actually not visible on the MCP panel. Instead the pilot formulates a mental representation of the goal state and monitors the progression of the aircraft towards this goal. Therefore, basic kinematics and

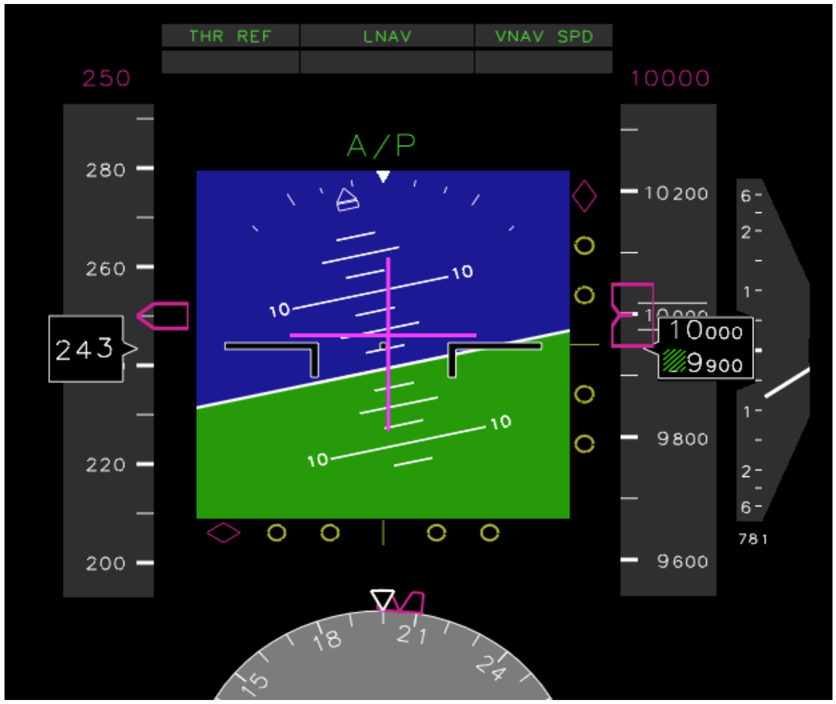

Figure 10: Principal Flight Display (PFD) used by pilots to monitor important states of the aircraft such as Airspeed(left tape), Altitude (right), attitude(center) and modes (top)

general knowledge of the modes will be part of the initial mental model of the pilots in the simulation.

CHAPTER IV

A COMPUTATIONAL APPROACH TO SITUATION AWARENESS

4.1 Situation Awareness

The concept of situation awareness (SA) was extensively developed by Endsley [8, 9] as an attempt to explain poor decision making and performance related to human interaction with complex dynamic systems. As theorized by Endsley, situation awareness not only describes the awareness of "numerous pieces of data" but also an "advanced level of understanding" as well as the "projection of future system states in the light of the operators goal." This characterization breaks SA into three levels (L1, L2, L3) respectively related to perception, comprehension and projection. However, the static knowledge of procedures and checklists does not fit well into this frame although they also support decision making. SA only gathers the dynamic factors of the immediate knowledge of the system.

4.1.1 Situation Awareness Level 1, Perception

SA L1 measures the perception of various elements of the environment. On the flight deck, these elements include instruments, external disturbances and their changing rate. For example, a pilot checking his altimeter and integrating visual and aural cues is an attempt to maintain L1 SA. The quality and comprehensiveness of the perception of relevant pieces of data is the first step towards a good situation awareness and the next levels strongly rely on it. An operator achieving a good SA L1 will have efficient and comprehensive monitoring patterns for relevant states and their history.

4.1.2 Situation Awareness Level 2, Comprehension

SA L2 exploits the data collected through SA L1 to provide an understanding of the current state. A good SA L1 has the pilot monitor his airspeed frequently during the final approach where SA L2 is his ability to turn this knowledge into identifying a close to overspeed or stalling situation. Also SA L2 allows the operator to infer certain states from the observation of others by the awareness of coupled dynamics and rules of the system. For example, if a pilot observes a positive vertical speed, he will expect the altitude to increase.

4.1.3 Situation Awareness Level 3, Projection

SA L3 corresponds to the projection of the knowledge acquired through L1 and L2 to a near future. Air traffic controllers aware of the current motion of multiple aircraft and understanding the common flight paths for that sector might be able to predict a future traffic congestion and act in advance to alleviate it. In a rapidly changing environment SA L3 is therefore crucial for efficient decision making. Moreover, SA L3 certainly relies on an accurate perception and understanding of the current state (L1 and L2).

4.2 A Computational approach

The three levels of *situation awareness* form an efficient base of projection of the operators understanding continuum as far as human-automation interaction is concerned. Methods like SAGAT [9] attempt to assess SA from HITL experiments but can lead to biased measurements as the operator is aware of being monitored. Moreover, running HITL experiments is expensive and needs a operational prototype of the system. Therefore, SA cannot be taken into account early in the design phase. Simulating SA through a computational approach would allow to use human agent models to detect earlier potential issues and dangerous situations due to a poor design or training. However, integrating this concept into a computational work modeling approach is not straightforward. As computer scientists still struggle at implementing the concept of human comprehension, this work does not ambition to propose a comprehensive computational model of human understanding but simply aims to model and implement constructs that account for parts of SA L1, L2 and L3. These constructs will be integrated in an advanced human agent model. They are independent from the work model where all knowledge of the work domain is encoded and thus can be applied to various kinds of systems. The implementation of *situation awareness* into a human agent model is a first necessary step to make simulations of human-automation interactions more realistic and diagnostic. This chapter describes the data structures required to simulate *situation awareness* within the agent model in WMC, the processes responsible for initializing and maintaining the different levels of SA will be described in the next chapter.

4.2.1 Perception and workload (SA Level 1)

Situation awareness Level 1 mostly describes the comprehensiveness and accuracy of the operator's perception of available cues about the state of the system [9, 8]. Did the driver see the car in his right side-mirror? Did the pilot hear the aural stall warning? Certain monitoring schemes such as the T-scan for pilots are part of the training and are performed on a recurrent basis depending of the available mental workload. If the pilot is otherwise engaged, e.g when talking on the radio or to his copilot, he might skip one of these monitoring actions and his SA will begin to degrade. Monitoring actions can be created within WMC as part as the work model and contribute to workload saturation. This way, the existing framework already serves the purpose of SA Level 1.

Situation awareness intrinsically conflicts with the work modeling paradigm described in the last chapter. Indeed it supposes that an instance of the agent model would carry some knowledge about the world whereas agents in WMC were initially designed to handle the execution of actions receiving their content from the work model implementation. Accordingly, new constructs were needed, this work introduces *mental resources* that are mental copies of the actual world *resources* and are updated by *monitoring actions*. This mental representation of the world is automatically generated as the agent experiences new monitoring actions and thus does not require a specialized agent model for a particular work model.

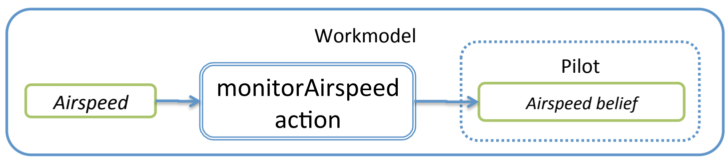

Figure 11: Symbolic representation of a work model with an Airspeed resource, the corresponding monitoring action that acts on an instance of a human agent model and updates its belief. The monitoring action "gets" the actual value of the airspeed and "sets" the belief in consequence.

4.2.2 Mental representation and comprehension (SA Level 2) : Bayesian Approach

4.2.2.1 Belief system

Situation Awareness Level 2 describes the comprehension of the perceived cues, the identification of variables of interest (close to dangerous boundaries for instance) necessary to feed the decision-making process in response to environmental inputs [9, 8]. An important aspect of SA L2 is to account for the operator's degree of belief of the state of the system as a consequence of the perception of numerous cues. Expert human agents are hypothesized to maintain a belief of the state of the system as well as a degree of confidence that can be high for variables directly monitored on trustful instruments and very low in case of inconsistent or out-of-date information. A novice operator is more likely to apply procedures and rules all the time whereas a more experienced agent will selectively consider risks and other contextual information whereas the expert will use his degree of confidence to reason with bayesian decision-making. Such an approach not only uses the last monitored value of a variable as a basis for decisions but also the degree of belief that one has about it.

To account for expert behaviors, we can introduce the *mental state* of the system as a set of probability distributions. Each of individual state probability distribution represents the comprehension of an actual variable of the system and is centered on the current belief of its value and stretched in or out as confidence decreases or increases.

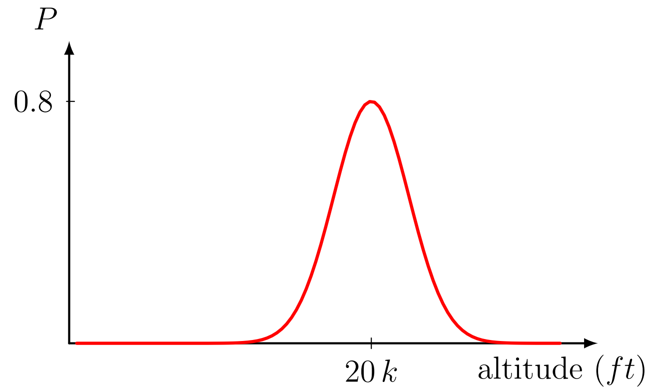

Figure 12: Example of mental state belief representing the altitude

Figure 12 shows an example of such a bayesian model for the internal representation of the pilot regarding the altitude. The connection between SA L1 and L2 is modeled by the impact of monitoring actions or the absence of up-to-date data from monitoring actions on the shape of the probability distribution of a given mental state. When the human agent monitors a state variable from an instrument, the mean of the probability distribution immediately shifts to the new observed value whereas the shape of the probability distribution narrows. For simplicity, will first assume normally distributed belief variables. The validity of such an assumption is discussed later in this chapter. Moreover, this work assumes that the operator trusts his instruments such that the consequence of a monitoring action on the gaussian belief probability is to set the mean to the observed value. Since we consider a gaussian distribution the shape of the believed variable is fully captured by the mean and the standard deviation. The consequence of the monitoring action is also to improve the confidence the operator has in his belief which will be computationally translated by bringing the standard deviation back to a reset value corresponding to the sensors accuracy or the reading error as illustrated in 13.

An important remark is that the belief distributions translates a bayesian rather than frequentist interpretation of probabilities. This means that the pilot does not sample from the distribution by accessing his working memory but simply reads the expected value, i.e the mean. The standard deviation does not represent his inability to remember or access to the actual value but his confidence in the value of this variable.

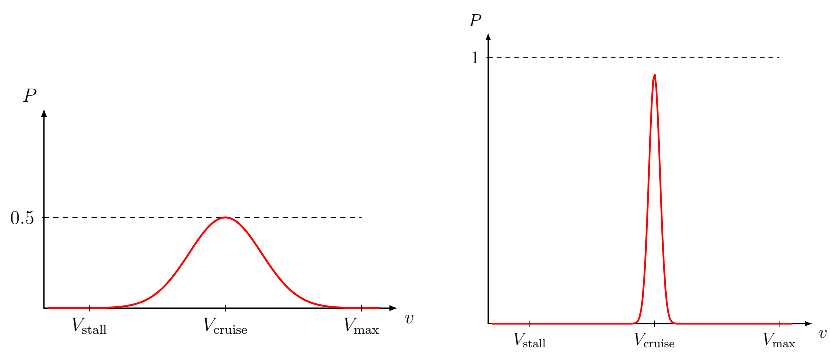

Figure 13: Example of mental state belief representing the speed before (left) and after(right) a monitoring action

4.2.2.2 Evolution of the belief between monitoring actions

Another interaction between SA L1 and L2 is the absence of monitoring action that decreases the degree of confidence of the operator and affects his decision making process. The process of decreasing confidence is continuous whereas the impact of events such as monitoring actions from instruments is discrete. Confidence in the knowledge of a state variable will decrease as a function of time and potentially of the knowledge of known boundaries of the derivative of the state.

4.2.2.3 Coupled dynamics

Such an implementation allows the operators belief of a certain state to be updated as a consequence of the observation of this state through monitoring actions. Another aspect of SA L2 is the understanding of how the value of a certain state, e.g vertical speed, impacts the rest of the system, e.g altitude. Therefore, the belief of a state variable system must be able to be updated by the perception of another state variable. The knowledge of specific dynamics coupling different variables is discussed in the following chapter about *mental models* as it is not part of the concept of SA but a process to maintain it. Figure 14 illustrates the evolution of the altitude belief state without monitoring but as the result of coupled dynamics and growing uncertainty.

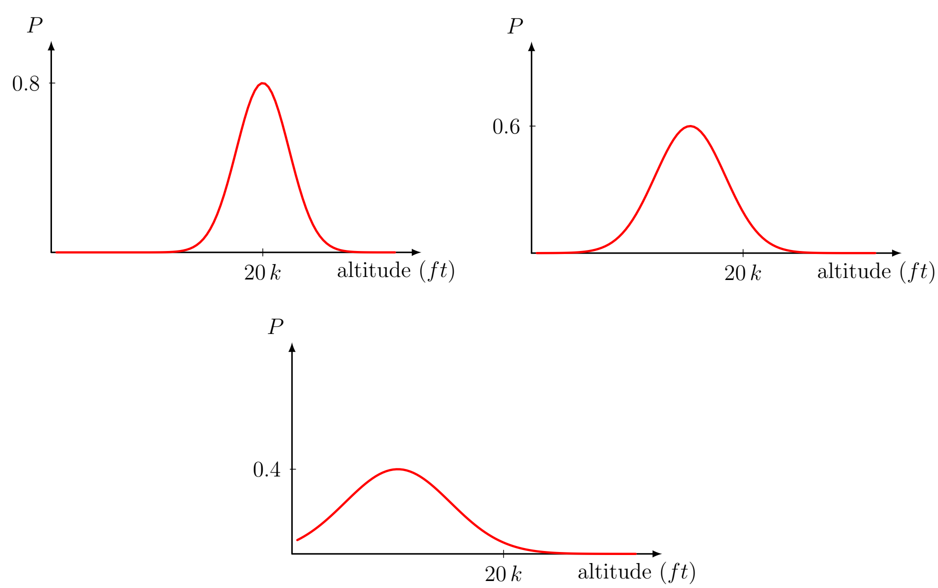

Figure 14: Altitude belief state as a result of the knowledge of a negative vertical speed and the absence of monitoring action

4.2.2.4 Gaussian assumption

We decided to project one's belief of a state to a two dimensional space (Current believed value, degree of confidence), assuming a gaussian distribution does not imply a loss of information. Moreover, using a gaussian representation has many advantages. First, it is easy to manipulate and memory-efficient. Indeed, we only need to store the mean and the standard deviation. Furthermore, this work will address mental models and model-based observers such as Kalman filter which assume normally distributed variables. We will discuss this assumption in section 4.4.

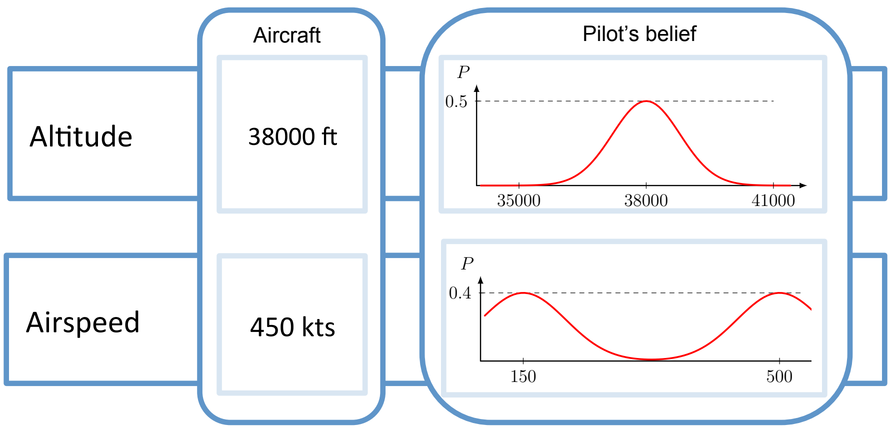

Figure 15: The operator's belief is generally centered on the actual value of the state it represents. When sensors are not available anymore, the belief can diverge from the actual value and even be multi-modal. In the AF447 accident case, the pilot did not know whether he was in an overspeed or stall situation.

4.3 Situation Awareness Metrics

Situation awareness is the interface between the operator's internal processes and the external world. Endsley's extensive theory of *SA*'s measurement techniques [9] provides insight into what could be good *situation awareness* metrics but also puts emphasis on the limitations of these techniques. Indeed, most of them involve subjective measures from subject matter experts participating in a simulation, which is randomly paused for assessment. This work models several aspects of *SA* that allows us to assess it in real time without interrupting the simulation as we define relevant metrics.

4.3.1 Standard deviation

As we stated in the background section of the introduction chapter, situation awareness is not totally assessed by the accuracy of the state's belief, i.e the mean of the state's density function but also through the standard deviation which described the confidence on the operator. Thus, a good indicator for instantaneous situation awareness is the breadth of the density function. The representation of *situation awareness* as a result of events such as monitoring actions is shown on figure 16. We see that monitoring events strongly decrease the uncertainty of the pilot whereas internal dynamics have the expected effect: increasing the uncertainty around an evolving expected value. Indirect dynamics represent the effect of the monitoring of coupled variables described in chapter 6. Here the monitoring of the vertical speed has an influence on the *situation awareness* of the altitude state.

4.3.2 Mean error

As Endsley [9] proposes to measure the operator's SA by assessing his knowledge of the world's state at discrete "freeze" events, we can continuously assess the discrepancy between the agent's belief of the world $\hat{x}$ and the actual state x. Therefore, we can introduce the total root mean square error as an aggregated metric to describe the overall situation awareness of a specific state over a certain amount of time:

$$E = \sqrt{\int_{t_0}^{t_1} (\hat{x} - x)^2} \tag{1}$$

and define SA_1 as a metric describing the accuracy of *SA L1/L2* during a specific time window as

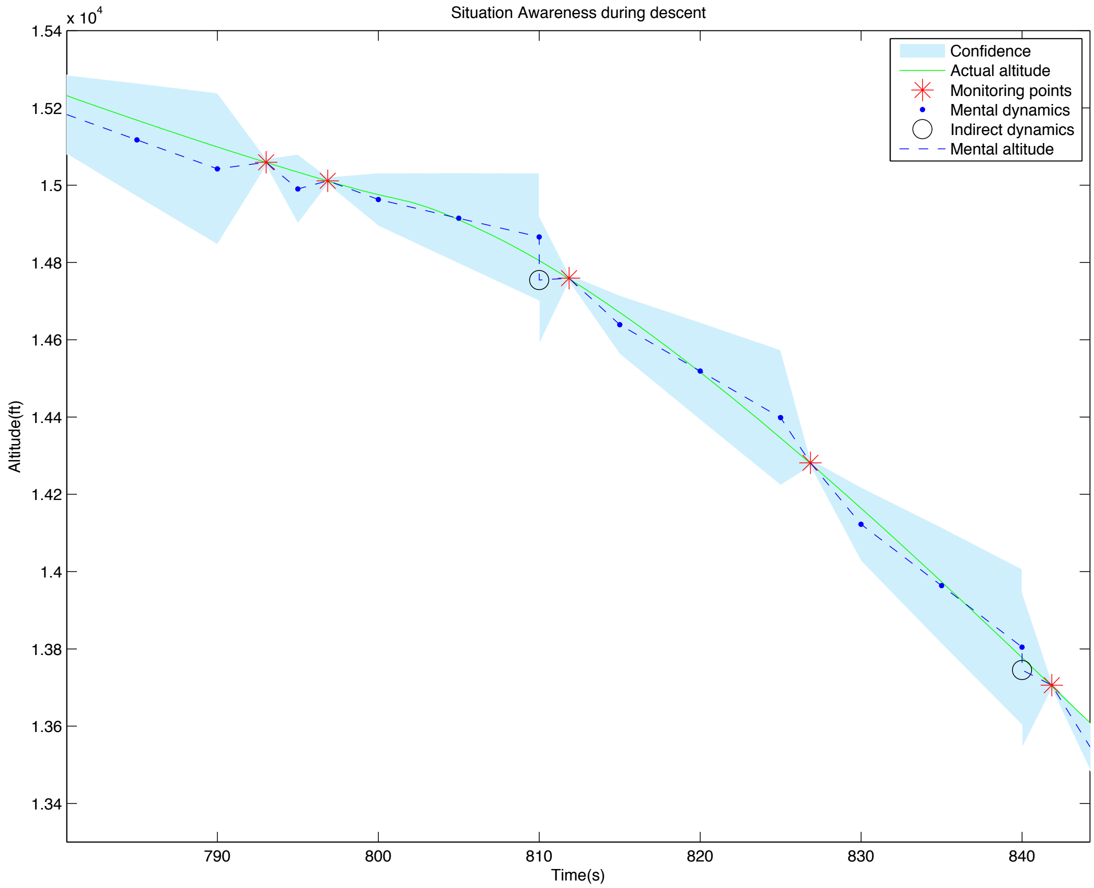

Figure 16: Evolution of SA during the approach phase of the test case scenario. 'Confidence' spans an area of three standard deviations below and above the mean value of the mental belief (mental altitude).

$$SA_1 = \frac{1}{E} = \frac{1}{\sqrt{\int_{t_0}^{t_1} (\hat{x} - x)^2}} \tag{2}$$

that goes to infinity when $x = \hat{x}$.

4.4 Non-gaussian distribution and cognitive dissonance

However, in some situations, human operators can have a multi-modal belief of certain variables that can lead to inconsistent actions or so-called cognitive dissonances. In the AF-447 accident report by the BEA, the experts conclude several times that the pilot flying hesitated between identifying an overspeed or stalling situation leading to non-consistent action sequences and eventually to the crash of the aircraft. In such a situation, the mental representation of the speed of the aircraft cannot be fully captured by a gaussian variable. Indeed bayesian decision making picks a decision policy that minimizes the risk with respect to the entire belief distribution. Allowing multi-modal probability distribution, like a sum of gaussian distributions can provide this capability while keeping a sparse belief representation as illustrated in Figure 15.

4.5 Summary

In this chapter we have introduced a simple probabilistic way of projecting *situation awareness* L1 and L2 onto a computational basis. We also identified the main processes responsible for updating SA such as monitoring actions, mental simulation of the system dynamics and the fact that confidence decreases between two monitoring events. The next chapters will describe these processes in details, how they interact with *situation awareness* and how this work implements them in the context of mental models.

CHAPTER V

SITUATION AWARENESS AND MENTAL MODELS OF DISCRETE DYNAMICS

As explained in chapter 4, *situation awareness* provides a frame to project the understanding of the operator on a computational basis. This section will outline the new constructs needed by the human agent model to update situation awareness over time as a result of discrete dynamics. The last part of this chapter also approaches the long term modification of discrete mental models as an extension of Javaux's work about implicit learning of autopilot mode transitions [16]. Discrete dynamics rule the behavior of discrete variables or modes such as `flap_setting` or `pitch_mode`. Such dynamics can be represented finite state machines with transition rules. A transition rule can be represented as a boolean expression where literals stand for conditions on state variables. Transition will occur if the resulting condition is satisfied. One of the contributions of this work is to predict dangerous consequences of high workload or lacunary mental models. This chapter proposes a computational approach to discrete mental models and integrates them into simulation. The plastic representation proposed by Javaux and extended in this work also allows the simulation to account for faulty mental models as a result of high workload or long term modifications.

5.1 A tree-based representation for discrete rules

Javaux's work illustrates how the pilot's knowledge of discrete rules such as automation mode transitions can impact situation awareness and therefore have to be part of the mental model [16]. An inaccurate knowledge of automatic mode transitions of the autopilot can lead to a problematic mode confusion. Transitions between modes invoke complex engagement conditions and some of them are met rarely enough that most pilots forget their importance [16]. To simulate this plasticity, we can model such mode transitions as a set of weighted conditions composing a boolean expression. The weights represent the perceived importance of each condition. The current workload establishes a threshold and conditions with lower weights are not retrieved into the operator's working memory [16], i.e the operator forgets these conditions.

Depending of the current workload, the operator is not likely to remember half-forgotten conditions [16]. Thus a threshold under which conditions with a low perceived importance will not be retrieved to working memory is set as a function of workload as illustrated in figure 18. A tree is a flexible data structure allowing the simulation tool to prune the boolean expression during a simulation run depending on the weights as shown in figure 19.

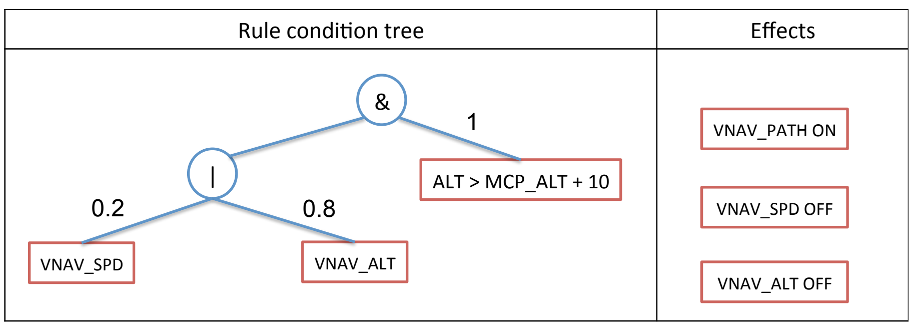

Figure 17: Example of a transition rule. & represents the operator AND and — stands for OR. VNAV_SPD, VNAV_ALT and VNAV_PATH are autopilot pitch modes, MCP means Mode Control Panel.

5.2 Mental model's plasticity or implicit learning

Initial training can be considered as the primary method to establish a good mental model in the agent. Although Doyle qualifies mental models as relatively enduring [6], they are not static. Experience will alter one's mental model and thus impact the

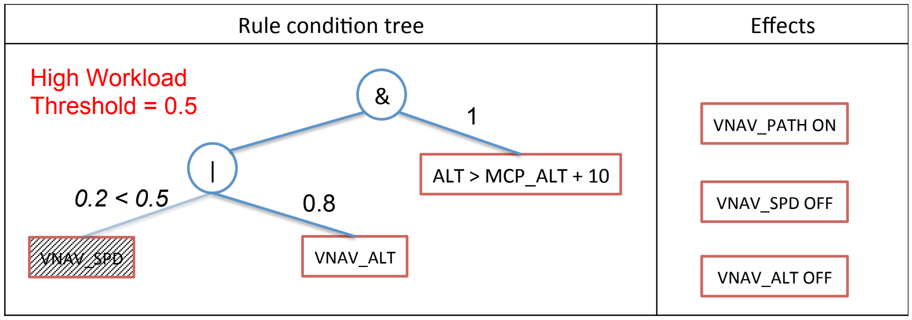

Figure 18: The same transition rule as in Figure 19 but with a high workload setting an arbitrary threshold to 0.5. The transition between VNAV_SPD and VNAV_PTH is not retrieved in this case.

maintenance of a good *situation awareness*. This section defines the long-term plasticity of *mental models* that makes them vulnerable to unconscious learning processes and describes how plasticity and learning are incorporated into the agent model.

5.2.1 Implicit learning of rules

Javaux's work [16] about how learning shapes the user's knowledge of a system explains the relatively poor knowledge of pilots when it comes to remembering transition rules between autopilot modes. He proposed using finite State Machines to formalize the modes and their transition conditions. A transition rule R_i is activated when a certain number of conditions C_j are satisfied. In order to implement implicit learning, Javaux introduces weights $w_{i,j}$ that represent the perceived importance of the condition C_j with respect to the rule R_i. Our work builds upon this idea using transition trees for computational simplicity.

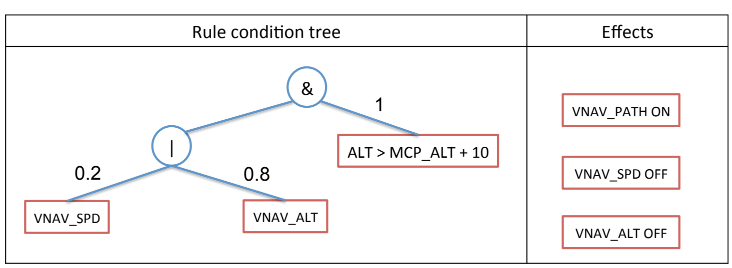

Figure 19: Example of a transition tree.

This alteration of knowledge can be captured by the concept of hebbian learning [15, 2, 5] based on the Hebb's rule known as a reference to explain synaptic plasticity in neuroscience, stated by Donald Hebb in 1949 and often summarized as "Cells that fire together, wire together". A general formula representing hebbian learning is:

$$\Delta w_i = \eta x_i y \tag{3}$$

where x_i represents an input signal, y the output signal and w_i the strength of the synapse or connection between x_i and y. However, his equation is almost never used in this form because unstable by nature. Moreover this equation holds for neurons that continuously integrate the inputs. Therefore we need to adapt it for discrete transition scenarios with complex condition trees. The input signal x_i of a *discrete rule* is the state of the conditions (true or false) that has to be satisfied to trigger the transition. The output signal y is binary: either the conditions are satisfied and the transition is triggered ($y = true$) or not ($y = false$).

For computational reasons we want the weights w_i to stay between 0 and 1 so we added the stabilizing factors $1 - w_i$ and w_i in the following equations:

$$\Delta w_i = \eta \times h(x_i, y) \tag{4}$$

where function h is described in Table 5.2.1 with

$$\begin{cases} h_+ = (1 - w_i) \\ h_- = -w_i \end{cases} \tag{5}$$

Assuming a perfect initial training, weights w_i start with a value of 1. Then any time the transition scenario, i.e condition tree is queried, an increment of h_+ will

parent node	x_i	y	h
AND	True	True	h_+
AND	True	False	h_-
AND	False	False	h_+
AND	False	True	h_-
OR	True	True	h_+
OR	True	False	h_-
OR	False	False	h_+
OR	False	True	h_-

Figure 20: Mapping of the hebbian function h with respect to x_i an input condition, *parent* the operator overarching x_i and y the overall response of the condition tree.

be assigned to conditions that played or may have played an active role in either activating or inhibiting the transition. For example, a false condition under an AND node will be partially or fully responsible for not triggering the transition whereas if the same condition that is always satisfied will be negatively impacted if it rarely triggers the transition. This would be the case for safety conditions such as the bank angle of the aircraft between -15 deg and 15 deg that tends to be forgotten by pilots.

5.3 Test case

5.3.1 Evolution of weights

Long term modifications of mental models due to operational practices, i.e frequencies of occurrence of transition scenarios between different autopilot modes are believed to be captured by Hebbian learning as described in section 5.2. Figure 22 shows the evolution of weights for four conditions c_1, c_2, c_3, c_4 involved in the transition scenario described in Figure 21 with different frequencies following the rule described in table 5.2.1.

We observe that the weight related to condition C_4, i.e w_4 is constant and equals 1 although the frequency of C4 being true is 83%. This is explained by the fact that C4 is always crucial in the final result of the transition rule. If C4 is true, then the rule is satisfied, otherwise, the rule is not satisfied. This statement does not hold for

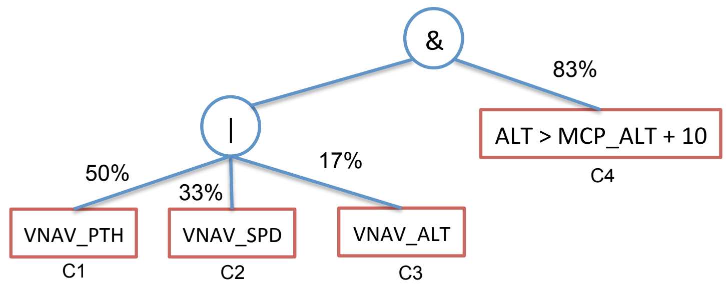

Figure 21: Example of transition rule with the frequencies at which conditions are met. These frequencies were arbitrarily derived from pilot common practices described in the B747 Training manual.

conditions C1, C2 and C3 which are child nodes of an OR operator. Therefore, their perceived importance decreases at a rate depending on their frequency, following the Hebbian rule described earlier in this chapter.

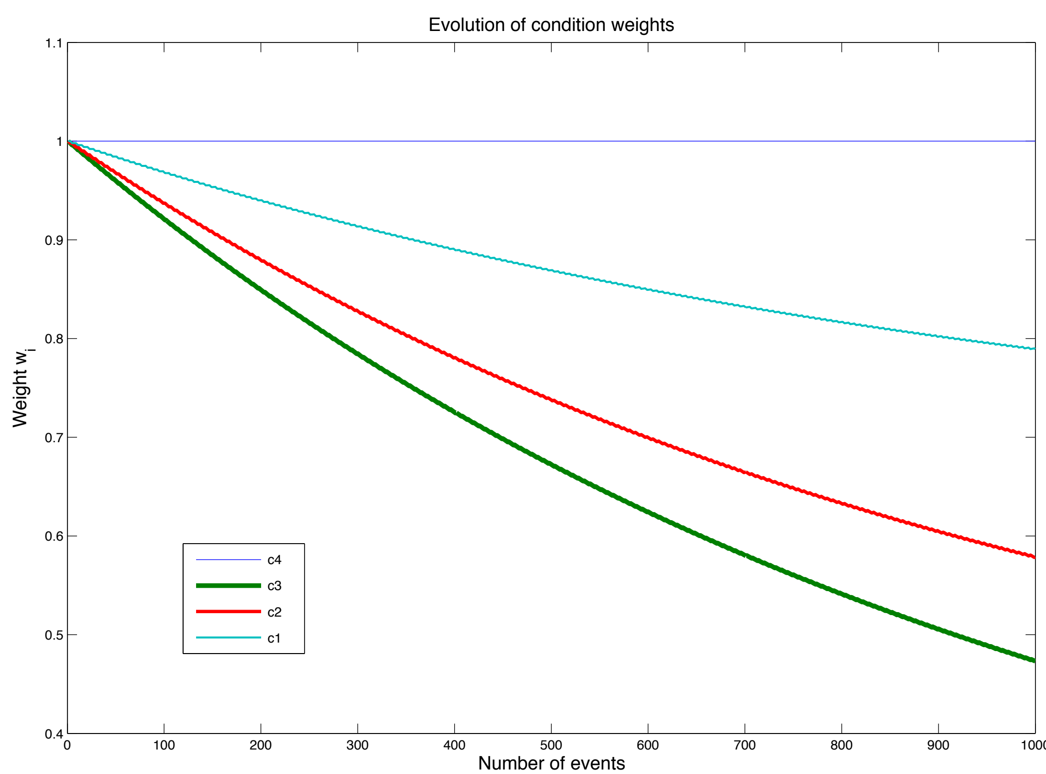

Figure 22: Evolution of the weights for the conditions of the rule described in Figure 21.

.

As the workload increases, the threshold required on the weight for a condition to be retrieved increases. Consequently, conditions C2 and C3 will be more likely to be forgotten. This process was implemented in WMC and used as a proof of concept for this example. Using realistic transition frequency data and running thousands of simulations allow the modeler to study the alteration of mental models, the consequences on *situation awareness* and the possible emergence of automation surprises.

CHAPTER VI

SITUATION AWARENESS AND MENTAL MODELS FOR CONTINUOUS DYNAMICS

Although authors like Forrester point out the dynamical limitations of mental models, the literature tends to show that agent models implementing model-based observers are able to match observations of expert operators like pilots [19]. Therefore, it is reasonable to assume that pilots run mental dynamic simulations to internally formulate their expectations regarding the evolution of the system's state. In situations where monitoring systems are faulty, mental models are the only way to maintain a reasonable belief of certain states. In the presence of observable states, a good accuracy of mental simulation of continuous dynamics allows humans to lower the monitoring frequency and reduce the mental and physical task load. This chapter introduces a formal way of simulating continuous mental dynamics (predict step) along with monitoring events (update step) based on model-based estimation theory such as Kalman filters.

6.1 Continuous Dynamics Actions

WMC is a continuous-time simulation engine. Its therefore capable of integrating numerically differential equations allowing to simulate complex non-linear system dynamics. These actions are executed by a non-human agent and generally include controllers and can implement a variety of numerical integration methods such as Runge Kutta. The mental version of *continuous dynamics actions* is called *mental dynamics* and can be either automatically generated from real dynamics or implemented manually. When a human agent executes a *mental dynamics action*, it gets the values of mental resources corresponding to the actual state variables involved and sets them to the new predicted value, as in equation 16 before updating the covariance matrix as described in the following section.

6.2 Mental dynamics

Mental dynamics can be approximated using the same mathematical formalism as actual dynamics. Let x be the actual state of the world, and $\hat{x}$ the belief, the basis for the operator's situation awareness. System dynamics ruling the world can be considered as a closed loop non linear system such as

$$\dot{x} = f(x) \tag{6}$$

Mental dynamics are likely to only act on a subpart of the state's estimation of the world. Let D denote a specific mental dynamical system acting on $\{\hat{x}^i\}_{i \in S_D}$, subset S_D of the whole mental state estimate $\hat{x}$. Since this work is supposed to serve numerical simulations, we will use a discrete system dynamics formalism, where the mental state at time $k+1$ is a function of the mental state at time k.

$$\hat{x}_{k+1} = f(\hat{x}_k) \tag{7}$$

At this point, situation awareness is represented in the fact that $\hat{x}^i$, the mental representation of the state x_i, is a gaussian probability distribution. This is compatible with such dynamics by maintaining a covariance matrix P that is updated at each iteration. The diagonal represents the standard deviation of the belief introduced in Chapter 4 and the off-diagonal elements measure the covariance between state variables, i.e how much the agent believes two states are linked by the system dynamics.

In mathematical terms :

$$P_k^{ij} = E[(x_k^i - \hat{x}_k^i)(x_k^j - \hat{x}_k^j)] \tag{8}$$

In particular,

$$P_k^{ii} = E[(x_k^i - \hat{x}_k^i)^2] = var(\hat{x}_k^i) \tag{9}$$

since $E(\hat{x}_k) = E(x_k)$.

The covariance matrix gets updated by the mental dynamics and more specifically by the Jacobian matrix of the dynamics:

$$F^{ij} = \frac{\partial f^i}{\partial x^j} \tag{10}$$

$$P_{k+1} = F_k P_k F_k^T \tag{11}$$

F^{ij} is computed numerically from the nonlinear mental dynamics at each iteration using Newton's central differences.

We can then derive the formula we will use in our implementation.

$$P_{k+1} = F_k P_k F_k^T \tag{12}$$

$$\rightarrow P_{k+1}^{ij} = \sum_m \frac{\partial f^i}{\partial x^m} (\sum_k P^{mk} \frac{\partial f^j}{\partial x^k}) \tag{13}$$

6.3 Human model-based observer

Kalman filters have been used extensively as an efficient way to model human estimation as part of Optimal Control Models [19, 20]. This section describes how this work uses Kalman filters to integrate monitoring actions with mental dynamics to simulate the impact of continuous mental models on *situation awareness*. Let D represent a dynamical system acting on a subset $\{\hat{x}^i\}_{i \in S_D}$ of the whole mental system's state $\hat{x}$ which is an estimator of the actual system's state x. $\hat{x}^i$ can represent the mental estimation of any continuous variable such as the *altitude* of the *airspeed* of the plane. Dynamics involved in aircraft models are nonlinear and represented by the function f. We will assume that observations z are the result of the agent directly monitoring state variables. Let P be the covariance matrix and $F = \frac{\partial f}{\partial x}$ Moreover, we are in the context of numerical simulations, therefore we will use the following mathematical discrete formulation. C represents the monitoring scheme, i.e the subpart of the state that is observed at iteration k.

$$\hat{x}_k = f(\hat{x}_{k-1}) \tag{14}$$

$$z_k = Cx_k \tag{15}$$

where z_k represents the observation from a monitoring action. It is important to note that steps 14 and 15 are asynchronous and generally do not happen at the same time.

6.3.0.1 Predict phase

The *predict* phase of the Kalman filter represents the simulation step. The mental estimation is updated through the mental non-linear dynamics and the covariance matrix is updated with respect to the following formula:

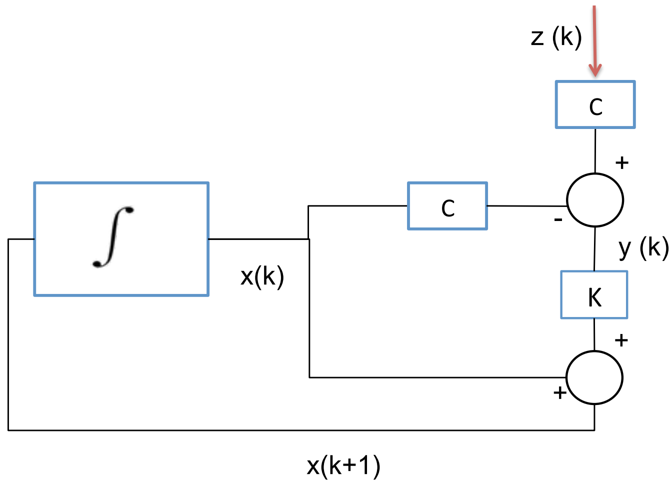

Figure 23: Control diagram representing the Kalman estimator

$$\hat{x}_k = f(\hat{x}_{k-1}) \tag{16}$$

$$P_k = F_{k-1} P_{k-1} F_{k-1}^T \tag{17}$$

The diagonal elements represent the uncertainty in the belief of each of the state. The values of the diagonal elements are increased by equation 17 as a consequence of integrating dynamics from uncertain state values.

6.3.0.2 Update phase

The update phase of the Kalman estimation integrates the direct observations from instruments into the mental belief and can be decomposed this way:

$$y_k = z_k - C\hat{x}_k \tag{18}$$

$$S_k = CP_kC^T + R_k \tag{19}$$

$$K_k = P_kCS_K^{-1} \tag{20}$$

$$\hat{x}_{k+1} = \hat{x}_k + K_ky_k \tag{21}$$

$$P_{k+1} = (I - K_kC)P_k \tag{22}$$

y_k in Equation 18 represents the difference between the monitored value and the current estimate of the state. Equations 19 and 20 compute the optimal gain K_k that allows Equation 21 to integrate the last observation in the next estimated state assuming neither process nor sensor noise.

Let us assume that the operator monitors one variable at a time. Then $C = [0...1..0]$ with the only non-zero element in the i-th column and $C\hat{x}_k = \hat{x}_k^i$. The equations can then be reduced to

$$y_k = z_k^i - \hat{x}_k^i \tag{23}$$

$$S_k = P_{ii} \tag{24}$$

$$K_k = \frac{1}{P_{ii}}Col_i(P_k) \tag{25}$$

$$\hat{x}_{k+1} = \hat{x}_k + K_ky_k \tag{26}$$

$$P_{k+1} = (I - K_kC)P_k \tag{27}$$

which can be simplified to

$$K_k = \frac{1}{P_k^{ii}} Col_i(P_k) \tag{28}$$

$$\hat{x}_{k+1} = \hat{x}_k + \frac{1}{P_k^{ii}} Col_i(P_k)(z_k^i - \hat{x}_k^i) \tag{29}$$

$$P_{k+1} = P_k - \frac{1}{P_k^{ii}} Col_i(P_k) Row_i(P_k) \tag{30}$$

The update of a specific element P^{ij} of the covariance matrix is computed by

$$\Delta P^{mn} = -\frac{1}{P^{ii}} P^{mi} P^{in} \tag{31}$$

Equation (29) shows that the specific monitoring of variable $\hat{x}^i$ can impact the whole mental state $\hat{x}$ as a consequence of non-zero off-diagonal elements of the covariance matrix, monitoring airspeed also updates estimates of altitude position. Moreover, some elements of this matrix are reset to zero in the absence of process and sensor noise. In order to maintain the filter's stability, we implemented a minimum standard deviation, corresponding to non-zero diagonal elements that represents the believed inaccuracy of the instruments or lack of confidence.

6.4 Test Case

The agent model created by this work was successfully implemented an used as the pilot in the described aircraft model. The pilot's initial mental model is comprised of simple kinematic dynamics k_1.

$$\begin{cases} h(k+1) = h(k) + v_s(k) \times \Delta t \\ v_s(k+1) = v_s(k) \end{cases} \tag{32}$$

The vertical speed V_s is monitored by the pilot every 30 seconds whereas the altitude h is monitored every 20 seconds plus as a reaction to certain events such as waypoint passing. Figure 24 shows the evolution of the mental belief of altitude (dotted line) as a result of three processes: direct monitoring of the altitude (stars), internal dynamics (k_1 applied to mental altitude and vertical speed, and indirect observation: the observation of the vertical speed tape also updates the belief of the altitude through the covariance matrix, as shown by equation 30 in Section 6.3. Kalman estimation is used to simulate the integration of all these information.

From this figure, it is clear that within legs of the descent profile with a constant vertical speed, the mental state is fairly close to the actual state. SA starts to degrade in areas of rapid or unexpected changes in vertical speed due to the behavior of autopilot control algorithms as seen between $t = 925$ and $t = 1010$ seconds. As the actual execution of mental dynamics actions and monitoring actions depend on the degree of vacuity of the pilot's active action list, i.e the current workload, we have to mention that figure 24 was obtained with WMC using the new agent model with a mental capacity larger than the biggest workload actually experienced during the descent, i.e the agent model never has to defer or interrupt actions.

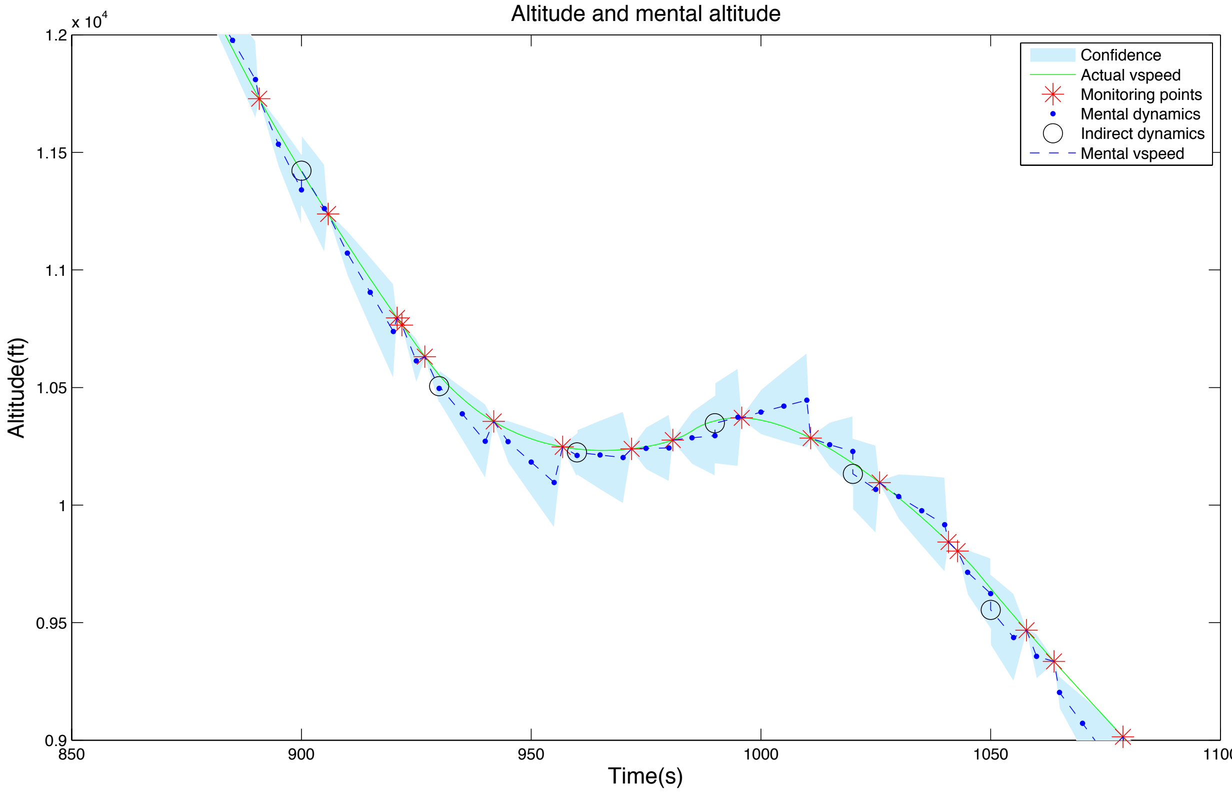

Figure 24: Evolution of the mental altitude as a result of observations, direct and indirect dynamic estimations. Here, indirect dynamics are the consequence of the observation of the vertical speed. The blue area represents an the uncertainty of the belief as much as three standard deviations below and above the mean

6.5 Measuring situation awareness

In Figure 24, a drop of *situation awareness* is observed between 925 and 1010 seconds, as the aircraft levels off automatically. Indeed, the estimated altitude deviates from the actual altitude. The overall situation awareness during the approach phase can be characterized by computing SA_1 metric.

To illustrate the impact of workload on SA, the mental capacity of the pilot was progressively decreased from 50 to reasonable values below 10 and Situation Awareness was measured as described in Section 4.3. Figure 25 gathers the results for several states of the aircraft. SA decreases with the mental capacity and we can observe a plateau for every state variable after 9 simultaneous tasks, which is the maximal workload any agent can experience from this *work model*. The y axis has been normalized between 0 and 1 for a better visualization.

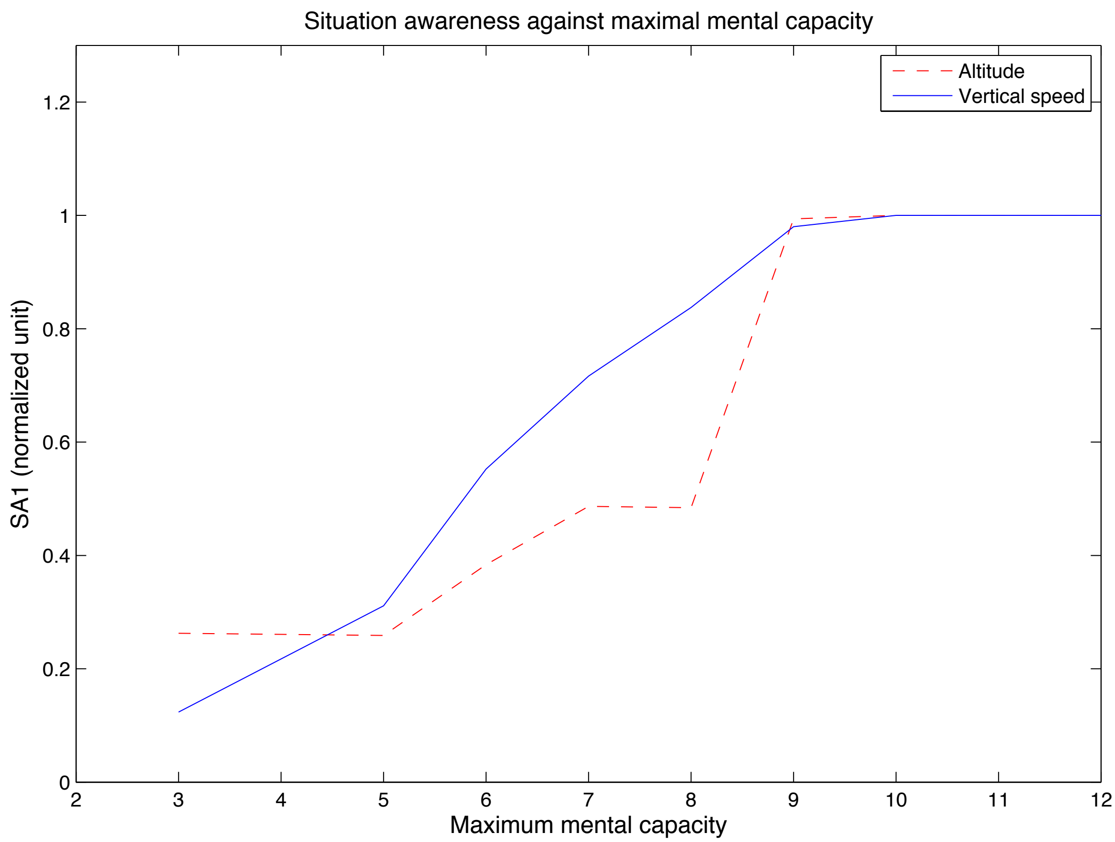

Figure 25: Evolution of SA_1 as a function of the mental capacity. Data has been normalized for each state variable so that the maximal root mean square error equals 1.

SA drops when the agent's active action list is saturated. Therefore, actions with lower priority are delayed or skipped. Considering the fact that humans are believed to struggle at handling multivariate dynamics, indirect dynamics are set to have the lowest priority, below direct dynamics and monitoring actions.The consequence can be seen in Figure 26, where more than 60 % of indirect observations are not accounted for when the scenario is run with a low agent's mental capacity.

This shows that this work provides a quantitative way of measuring the impact of too high workload or the loss of an instrument on situation awareness. In the case of AF447, such a simulation running live in the cockpit fed with eye-tracking observation data could have detected a drop in situation awareness. A dynamic display could then use this information to adapt the saliency of different instruments to try to restore the pilots' SA. On a design perspective, an engineer could simulate the drops in SA as a consequence of high workload and system complexity.

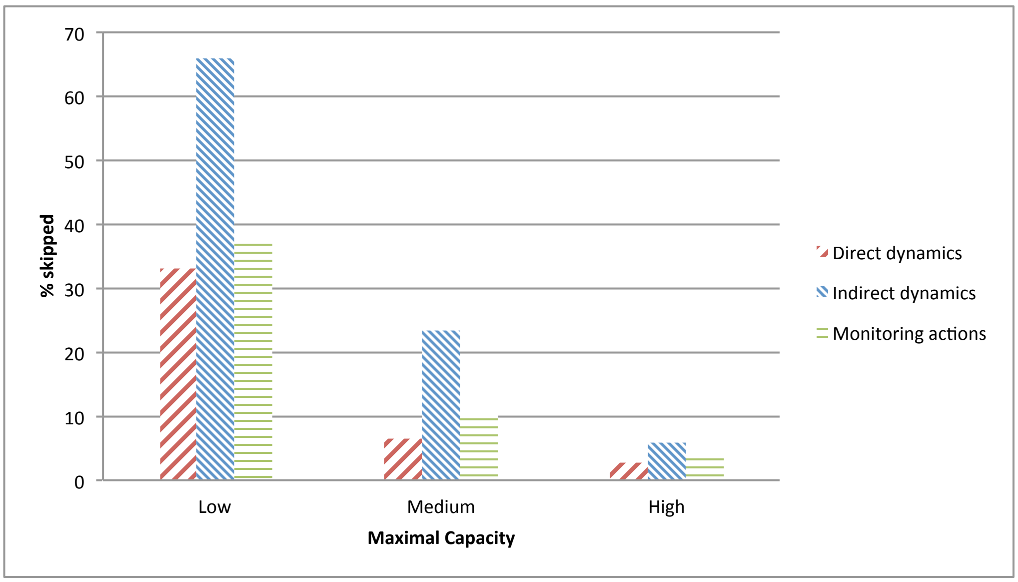

Figure 26: Pourcentage skipped per type for different mental capacities. (Low: 3, Medium: 5, High: 7) relative to the estimation of the altitude in the continuous approach scenario.

CHAPTER VII

MULTI-LEVEL APPROACH TO MODELING SOCIO-TECHNICAL SYSTEMS IN WMC 4

This work has defined a methodology to translate cognitive engineering concepts such as *mental model* and *situation awareness* into computational models. To allow engineers to actually use these models to support their design and analysis, we also focused on redesigning the base framework WMC in a more flexible and modular as well as user friendly fashion. This chapter describes shortly the new skeleton of work modeling in WMC 4 and how it improves the overall design, simulation and evaluation processes.

7.1 Toward a graphical integrated work modeling system

Stating the fact that work modelers are not necessarily experienced developers willing to interact with the heavy C++ machinery of WMC's simulation core, we intended to separate the modeling task from the simulation framework development. Before this work, WMC users would have to localize, understand and modify the C++ code responsible for defining parameters such as function allocation, scenarios, control algorithms with a high risk of negative interference with the simulation core itself. This new separation allows researchers to intervene at different levels of the system and to have a clear view on the design variables they are interacting with as shown in Figure 27.

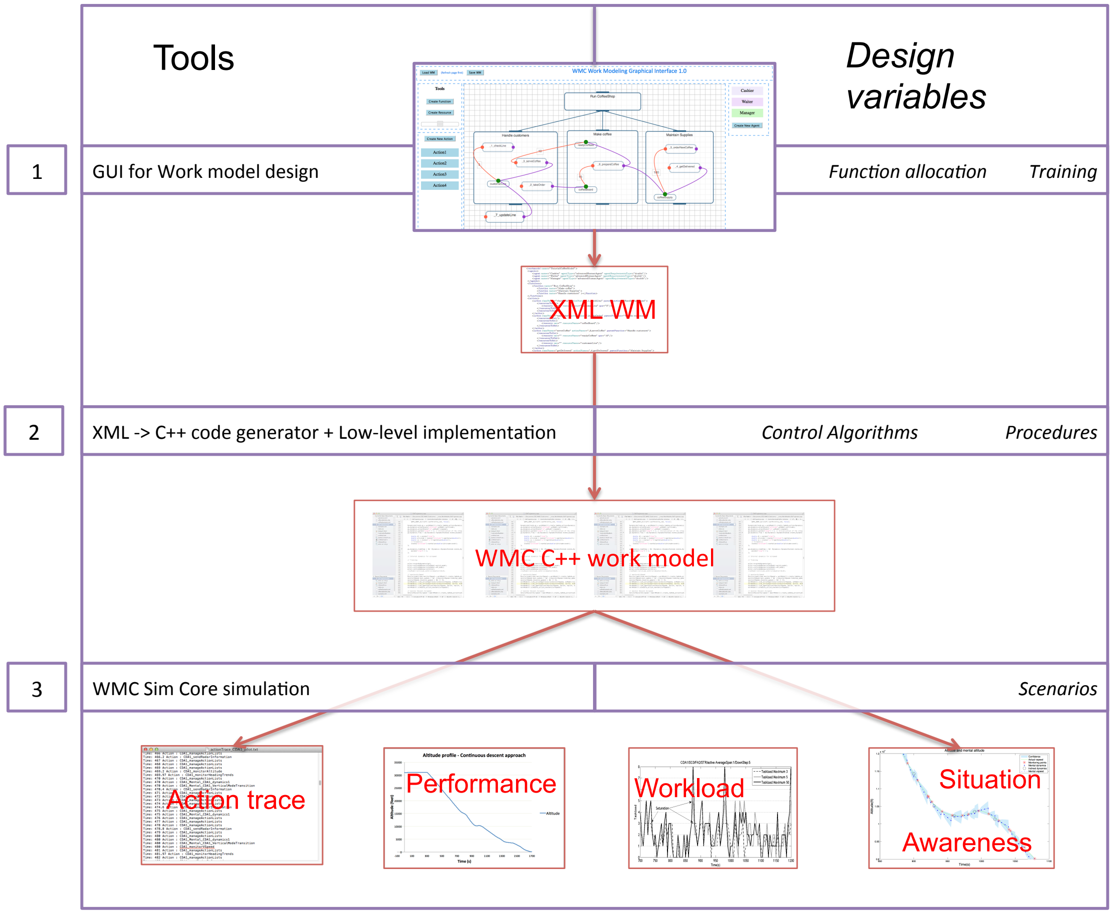

Figure 27: Overview of WMC 4 tools and design variables

7.1.1 Redefining workmodels

In practice, a WMC work model is a large `.cpp` file with everything inside from the definition of the abstraction hierarchy, resources and agents to the detailed implementation of every action. We decided to bring work models back to their core definition, i.e the abstraction hierarchy introduced in Chapter 2. It should be an abstract representation of the work easily modifiable for functional analysis rather than sequential code. The graphical representation is also essential as it supports multi-level functional modeling which was nearly impossible by the current sequential programming method.

7.1.2 XML and GUI

These requirements fit the standard format XML well. It is both human-readable and machine-readable and interacts easily with Graphical User Interface (GUI) using a Document Object Model (DOM) application programming interface (API) for instance. An example of such a work model representation is given in figure 29. Therefore, it is now possible to utilize a web-based user interface allowing the user to create and edit a work model, resources and actions graphically through a regular web browser instead of writing raw XML. Using the graphical interface, system designers do not need advanced programming skills anymore to use WMC and evaluate their design choices. It is now possible to compose work models and to change the function allocation across agents, the team composition in a few seconds. The GUI is able to generate an XML work model respecting a standard WMC grammar. WMC can generate C++ code out of the XML work model and placeholders for specific action implementations such as control laws or complex procedures leading faster to a C++ work model ready for simulations.

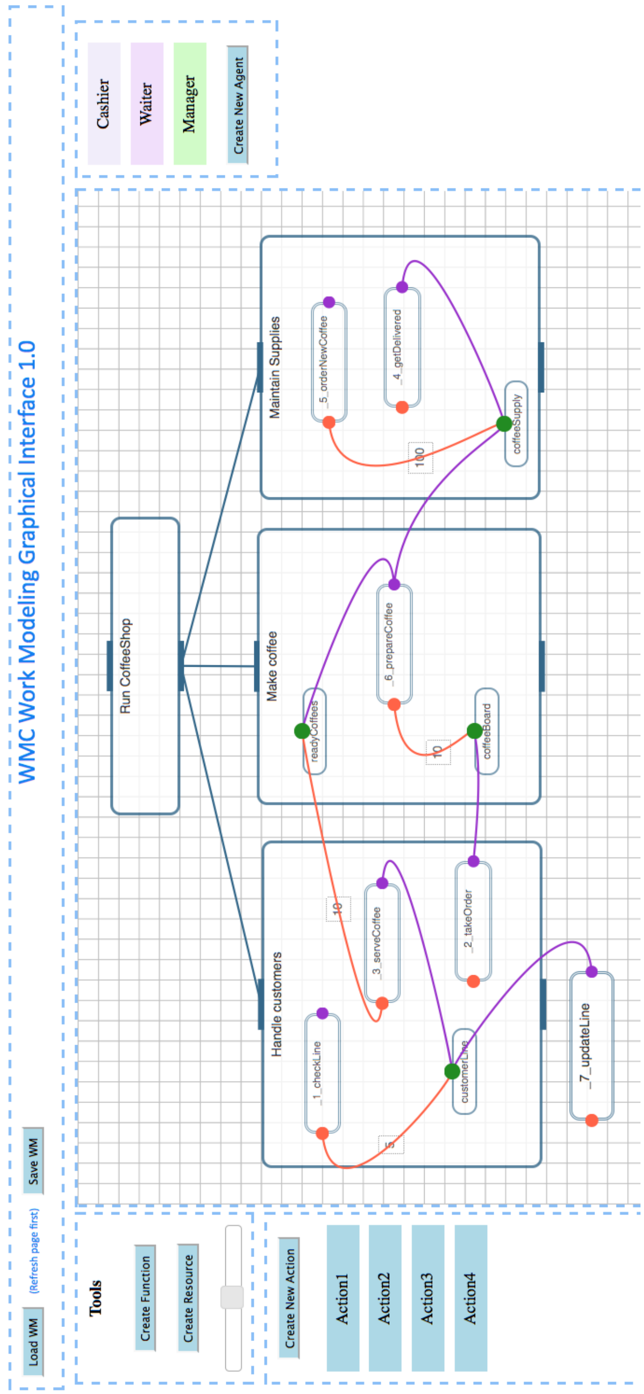

Figure 28: Web-based work modeling interface - Example of a simple coffee shop model

7.1.3 System Overview and future work

The overall system allows WMC's user to act upon three different levels represented in figure 27 :

1. *Work model design* : the GUI allows the user to create and modify the following:
   - Functional design
   - Function allocation
   - Agent's training (future work)

2. *Low-level implementation* : the code generator prepares the C++ files where the user can enter custom codes to implement specific :
   - Operational procedures
   - Automation-related algorithms (controls, modes)
   - System dynamics

3. *Simulation* : WMC Simulation core allows the user to define:
   - Scripted events
   - Designs of Experiment (future work)
   - Metrics

Finally WMC now produces standardized outputs such as sequential action traces, workload graphs for each agent as well as the evolution of situation awareness for every resource and agent. The user can also easily define and output performance metrics and plot them. The system overview proposed in Figure 27 demonstrates the achievement of several goals of this work. First it shows that human-related critical variables such as function allocation or training are now introduced early in the design phase. Moreover, situation awareness is an indicator that depends on training, function allocation as well as system dynamics. Therefore, WMC now allows to study interactions between human-related and automation-related independent variables and can therefore be used for quantitative HCI analysis.

```xml
<workmodel name="TutorialCoffeeModel">
 <agents>
     <agent name="Cashier" agentType="advancedHumanAgent" agentRequirementsType="double"/>
     <agent name="Waiter" agentType="advancedHumanAgent" agentRequirementsType="double"/>
     <agent name="Manager" agentType="advancedHumanAgent" agentRequirementsType="double"/>
 </agents>
 <functions>
     <function name="Run CoffeeShop">
         <function name="Make coffee">
         <function name="Maintain Supplies">
         <function name="Handle customers" ></function>
 </functions>
 <actions>
     <action className="checkLine" actionName="_1_checkLine" parentFunction="Handle customers">
         <resourcesToGet>
             <resource var="" resourceName="customerLine" qos="5"/>
         </resourcesToGet>
         <resourcesToSet/>
     </action>
     <action className="takeOrder" actionName="_2_takeOrder" parentFunction="Handle customers">
         <resourcesToGet/>
         <resourcesToSet>
             <resource var="" resourceName="coffeeBoard"/>
         </resourcesToSet>
     </action>
     <action className="serveCoffee" actionName="_3_serveCoffee" parentFunction="Handle customers">
         <resourcesToGet>
             <resource var="" resourceName="readyCoffees" qos="10"/>
         </resourcesToGet>
         <resourcesToSet>
             <resource var="" resourceName="customerLine"/>
         </resourcesToSet>
     </action>
     <action className="getDelivered" actionName="_4_getDelivered" parentFunction="Maintain Supplies">
         <resourcesToGet/>
         <resourcesToSet>
             <resource var="" resourceName="coffeeSupply"/>
         </resourcesToSet>
     </action>
     <action className="orderNewCoffee" actionName="_5_orderNewCoffee" parentFunction="Maintain Supplies">
         <resourcesToGet>
             <resource var="" resourceName="coffeeSupply" qos="100"/>
         </resourcesToGet>
         <resourcesToSet/>
     </action>
     <action className="prepareCoffee" actionName="_6_prepareCoffee" parentFunction="Make coffee">
         <resourcesToGet>
             <resource var="" resourceName="coffeeBoard" qos="10"/>
         </resourcesToGet>
         <resourcesToSet>
             <resource var="" resourceName="readyCoffees"/>
             <resource var="" resourceName="coffeeSupply"/>
         </resourcesToSet>
     </action>
     <action className="updateLine" actionName="_7_updateLine">
         <resourcesToGet/>
         <resourcesToSet>
             <resource var="" resourceName="customerLine"/>
         </resourcesToSet>
     </action>
 </actions>
 <resources>
     <resource name="customerLine" type="double"/>
     <resource name="coffeeBoard" type="double"/>
     <resource name="readyCoffees" type="double"/>
     <resource name="coffeeSupply" type="double"/>
 </resources>
</workmodel>
```

Figure 29: Example of an XML work model auto-generated through the GUI

CHAPTER VIII

CONCLUSION

This work introduced a quantitative and computational method of measuring situation awareness of pilots through simulating human agents' belief of the current state of the system and mental models of the dynamics. In Chapter 7, we explain how this development was integrated into an overarching design process and supports quantitative analysis of human automation interaction early in the design process.

8.1 More realistic simulations

However, SA is not simply a new metric but required significant additional simulation infrastructure. Moreover, accounting for it also changes the execution of the simulation itself. Indeed, the agent's decision making process is now fed with the agent's state belief as we can see in figure 30 and 31. Therefore, accounting for mental models and *situation awareness* make simulation more realistic and although further testing is still required to calibrate our system, this work opens the way to a new design method quantitatively integrating human factors as soon as the very first functional analysis. While, WMC does not aim to accurately simulate the complete response of pilots in the case of unexpected events but provides a comprehensive method for identifying the preconditions of poor human automation interaction. We believe this work can help defining new design envelopes relatively to HAI.

8.2 Hybrid mental dynamics

SA is updated through observation and two different types of mental models: discrete and continuous mental dynamics accounting for the pilot's perception of both discrete

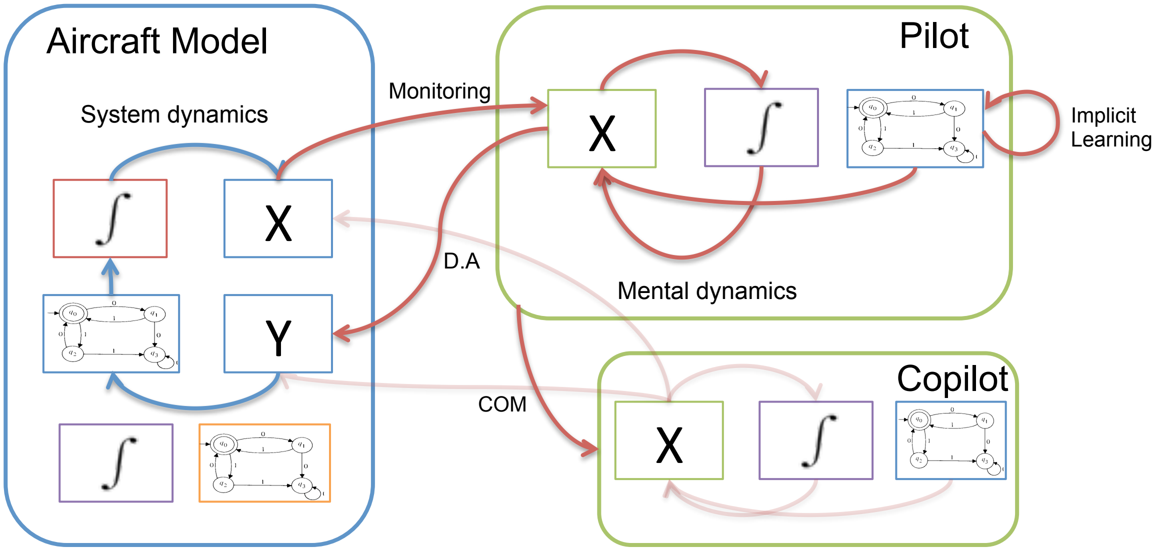

Figure 30: Overview of the simulation using the new agent model. The pilot monitors the system's state X and runs mental dynamics. He/she uses his/her comprehension of the world to take decisions and reconfigure the aircraft (Decision Actions, D.A) and also communicates with other agents (COM). Implicit learning shapes the alteration of discrete mental models.

dynamics such as autopilot modes and continuous dynamics, for instance, flight dynamics or complex control algorithms. Analysts can now study the pilot's interactions with discrete flight mode changes and long-term knowledge degradation phenomena as described in Chapter 5. It is also possible to study the effect of introducing a new control algorithm in the flight system along with different training strategies to measure the pilot's understanding and his situation awareness when facing an increasing complexity. More generally, this work allows engineers to simulate the interaction of hybrid - discrete and continuous - dynamics against the pilot's understanding and produce quantitative assessments of the resulting situation awareness.

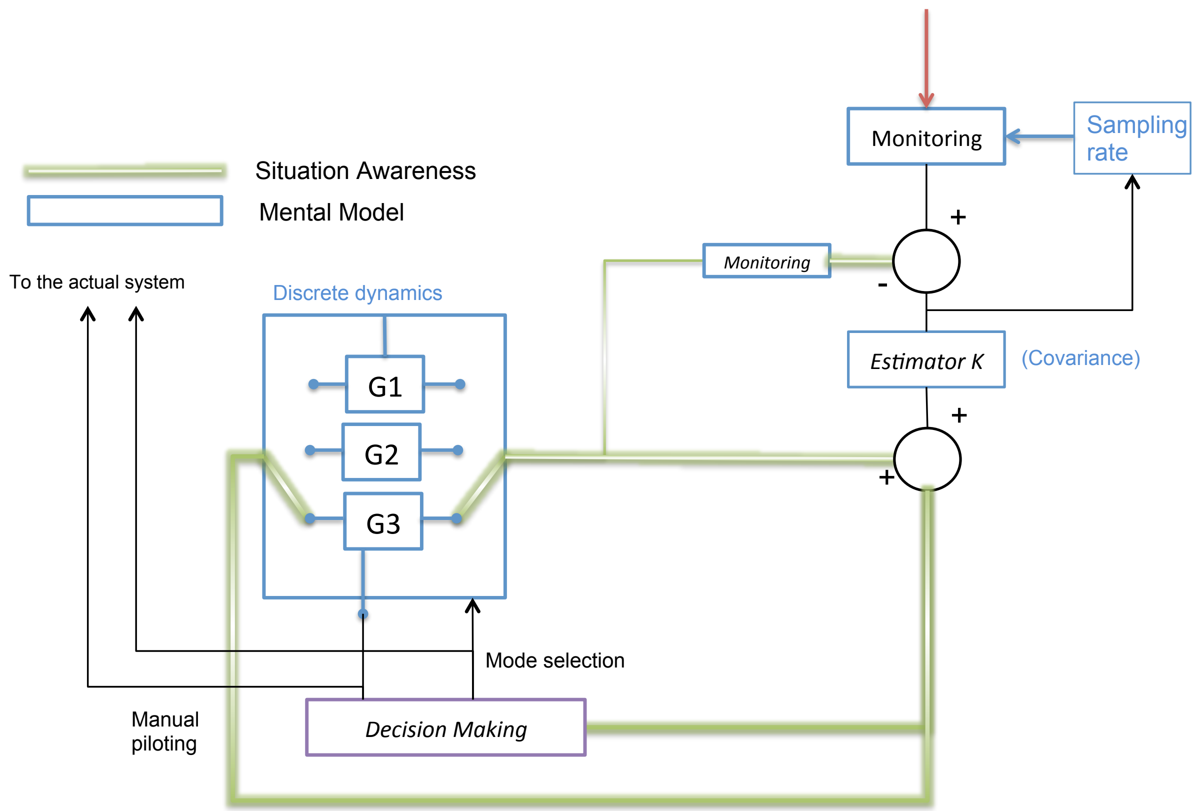

Figure 31: Overall system accounting for discrete and continuous dynamics. This diagram represents the estimated state as the major component of situation awareness whereas mental models are comprised of the dynamics and monitoring patterns. The decision making block shows how SA and mental models actually influence the actual system.

8.3 Application in Design and Certification

WMC is now a powerful design and analysis tool for socio-technical systems allowing engineers to work with design variables such as team composition, function allocation, training, automation algorithms, operational procedures, communication procedures, run thousands of scenarios and produce metrics such as user-define performance metrics, workload, situation awareness and long-term knowledge degradation. For instance, it is now possible to change the control laws of the autopilot and analyze the consequences on pilots' situation awareness. WMC can also be used to test whether additional training would improve the pilot's understanding or degrade his/her *situation awareness* as a result of added complexity. Accounting for *situation awareness* also allows accident investigators to simulate the loss of an instrument - for instance the speed indication in the case of AF 447 - and simulate the loss of SA whereas safety engineers could use WMC to test solutions to help pilots when these unexpected situations happen. A tool like WMC enhanced with the computational models developed in this work could also be a good start to develop quantitative validation metrics regarding the certification of automated systems in the context of human-computer interaction.

8.4 Real-time applications

Previous conclusions were centered on the positive impacts of this work on design. However, the *situation awareness* assessment system developed here could also provide a base for projecting real-time data onto a computational model of *situation awareness*. Thus, knowing the training of the pilot, and using for example live eye-tracking data, a continuous assessment of the pilot's SA could be made during the flight whereas a dynamic interface could adapt the saliency of certain information to prevent potential misunderstandings of drifts of SA. In the AF447 case, such a monitoring system would have tried to restore the pilots' SA by simulating their understanding of the situation and displaying relevant information such as a reminder of the switch from normal law to direct law.

8.5 Limitations

However, agent models are not likely to completely replace HITL experiments. Models cannot replace testing with actual pilots. The agent model developed in this work does not account for phenomena such as stress, confirmation bias or other psychological aspects that are known to play a major role in accidents. The new features of WMC's agent models do not aim to predict exact pilots' reactions and will not as a human model does not capture personality or tolerance to stress for instance. However, this work tries to approximate some aspects of pilots' experience through the hebbian learning theory developed in Chapter 5. This is a first step to specialize a general model to capture individual features.

8.6 Future work

To validate the models used in this work, and calibrate parameters such as the hebbian learning rate introduced in Chapter 5 about knowledge plasticity, we would need to run human experiments. After a robust calibration, we would need to determine safety envelopes for SA and progress toward introducing new certification metrics. A next step to improve the computational model of SA would be to account for SA L3, i.e simulate agents' anticipation. Indeed, when workload is low, pilots think ahead using mental models, seek information early and potentially take preventive decisions.

As far as aviation is concerned, the next step is to create relevant scenarios and improve the realism of aircraft work models to demonstrate the validation power of the method presented in this work. NASA recently pointed out serious safety issues in continuous descent approaches with pilots faced with unexpected speed restrictions. Implementing such scenarios in WMC will allow us to analyze related HAI issues and propose improvements in NextGen air traffic operations such as automation design, training and communication procedures.

8.7 Discussion

The fact that human automation interaction is not reducible to interface design but has to be analyzed at the system design level is still hardly understood in the industry. Human operators not only interact with automation through displays and actuators but also through their mental model of the system the accuracy of which highly depends on the complexity of the actual system and the operator's experience with it. We believe this work provides a solid first step toward providing a quantitative basis to promote the understanding and use of cognitive engineering in aviation and certification procedures.

REFERENCES

[1] Beyer, H. and Holtzblatt, K., "Contextual design," *interactions*, vol. 6, no. 1, pp. 32–42, 1999.

[2] Bi, G. and Poo, M., "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *The Journal of Neuroscience*, vol. 18, no. 24, pp. 10464–10472, 1998.

[3] Corker, K. M. and Smith, B. R., "An architecture and model for cognitive engineering simulation analysis: Application to advanced aviation automation," in *Proceedings of the AIAA Computing in Aerospace 9 Conference*, pp. 1079–1088, 1993.

[4] Crow, J., Javaux, D., and Rushby, J., "Models and mechanized methods that integrate human factors into automation design," in *International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000*, Citeseer, 2000.

[5] Dayan, P. and Abbott, L., "Theoretical neuroscience: computational and mathematical modeling of neural systems," *Philosophical Psychology*, vol. 15, no. 4, 2002.

[6] Doyle, J. and Ford, D., "Mental models concepts for system dynamics research," *System dynamics review*, vol. 14, no. 1, pp. 3–29, 1998.

[7] Endsley, M. R., "Measurement of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 65–84, 1995.

[8] Endsley, M. R., "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[9] Endsley, M., "Situation awareness global assessment technique (sagat)," in *Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National*, pp. 789–795, IEEE, 1988.

[10] Feigh, K. M., Gelman, G., Mamessier, S., and Pritchett, A. R., "Simulating first-principles models of situated human performance," *IEEE Transactions on System Man and Cybernetics: Part A.*, Submitted June 2012.

[11] Forrester, J., *Industrial dynamics*, vol. 2. MIT press Cambridge, MA, 1961.

[12] Forrester, J., "Counterintuitive behavior of social systems," *Theory and Decision*, vol. 2, no. 2, pp. 109–140, 1971.

[13] Forrester, J., "Policies, decisions and information sources for modeling," *European Journal of Operational Research*, vol. 59, no. 1, pp. 42–63, 1992.

[14] Galotti, K., "Approaches to studying formal and everyday reasoning.," *Psychological Bulletin*, vol. 105, no. 3, p. 331, 1989.

[15] Hebb, D. O., *The organization of behavior: A neuropsychological theory.* Lawrence Erlbaum, 2002.

[16] Javaux, D., "A method for predicting errors when interacting with finite state systems. how implicit learning shapes the user's knowledge of a system," *Reliability Engineering & System Safety*, vol. 75, no. 2, pp. 147–165, 2002.

[17] Javaux, D., "Explaining sarter and woods classical results," in *Second Workshop on Human Error, Safety, and Software Design, University of Washington, Seatle, WA*, Citeseer, 1998.

[18] Kim, S., "Model-based metrics of human-automation function allocation in complex work environments," 2011.

[19] Kleinman, D. L. and Curry, R. E., "Some new control theoretic models for human operator display monitoring," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 11, pp. 778–784, 1977.

[20] Kleinman, D. L. and Curry, R. E., "Some new control theoretic models for human operator display monitoring," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 11, pp. 778–784, 1977.

[21] Leveson, N., Pinnel, L., Sandys, S., Koga, S., and Reese, J., "Analyzing software specifications for mode confusion potential," in *Proceedings of a Workshop on Human Error and System Development*, pp. 132–146, Citeseer, 1997.

[22] Meadows, D., Meadows, D., Randers, J., and others, *Beyond the limits: global collapse or a sustainable future.* Earthscan Publications Ltd., 1992.

[23] Morecroft, J., "Executive knowledge, models and learning," *European Journal of Operational Research*, vol. 59, no. 1, pp. 9–27, 1992.

[24] Mosier, K. L. and McCauley, S. T., *Adaptive Perspectives on Human-Technology Interaction: Methods and Models for Cognitive Engineering and Human-Computer Interaction*, ch. Achieving Coherence: Meeting New Cognitive Demands in Technological Systems, pp. 163–174. Oxford University Press, 2006.

[25] Mosier, K. L. and Skitka, L. J., *Automation and Human Performance: Theory and Applications*, ch. Human Decision Makers and Automated Decision Aids: Made for each other? 1996.

[26] Norman, D., "Some observations on mentai modeis," *Mental models*, vol. 7, 1983.

[27] Norman, D., *The design of everyday things*. Basic books, 2002.

[28] Pritchett, A., Kim, S., Kannan, S., and Feigh, K., "Simulating situated work," in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pp. 66–73, IEEE, 2011.

[29] Rasmussen, J., "{Information Processing and Human-Machine Interaction. An Approach to Cognitive Engineering}," 1986.

[30] Rasmussen, J., Pejtersen, A. M., and Goodstein, L. P., *Cognitive systems engineering*. John Wiley & Sons, Inc., 1994.

[31] Richardson, G., Andersen, D., Maxwell, T., and Stewart, T., "Foundations of mental model research," in *Proceedings of the 1994 International System Dynamics Conference*, pp. 181–192, 1994.

[32] Richardson, G. and Pugh III, A., *Introduction to system dynamics modeling with DYNAMO*. Productivity Press Inc., 1981.

[33] Sarter, N. B. and Woods, D. D., "Situation awareness: A critical but ill-defined phenomenon," *The International Journal of Aviation Psychology*, vol. 1, no. 1, pp. 45–57, 1991.

[34] Sarter, N. B. and Woods, D. D., "Pilot interaction with cockpit automation: Operational experiences with the flight management system," *The International Journal of Aviation Psychology*, vol. 2, no. 4, pp. 303–321, 1992.

[35] Sarter, N. B. and Woods, D. D., "Pilot interaction with cockpit automation ii: An experimental study of pilots' model and awareness of the flight management system," *The International Journal of Aviation Psychology*, vol. 4, no. 1, pp. 1–28, 1994.

[36] Sterman, J., "Learning in and about complex systems," *System Dynamics Review*, vol. 10, no. 2-3, pp. 291–330, 1994.

[37] Vicente, K. J., "A few implications of an ecological approach to human factors.," 1995.

[38] Vicente, K., *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Lawrence Erlbaum, 1999.