# Cognitive Modelling of Complex Problem Solving Behaviour

Alberto De Obeso Orendain

Thesis submitted for the degree of Doctor of Philosophy in Informatics
University of Sussex
May 2014

# Declaration

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

**Signature**:.............................................

**UNIVERSITY OF SUSSEX**

Alberto De Obeso Orendain

Thesis submitted for the degree of Doctor of Philosophy in Informatics

# Cognitive Modelling of Complex Problem Solving Behaviour

## Summary

In the universe of problems humans face every day there is subset characterized by a salient dynamic component. The FireChief task (Omodei & Wearing 1995) is a fire-fighting computer simulation that can be characterized as the acquisition of interactive skills involving fast-paced actions cued by external information. This research describes the process followed to create a cognitive model of this complex dynamic task where full experimental control is not available. The cognitive model provides a detailed description of how cognition and perception interplay to produce the interactive skill of fighting the fire. Several artefacts were produced by this effort including a dynamic task fully compatible with ACT-R, a tool for analysing the data, and a cognitive model whose features enable the replication of several aspects of the empirical data. A key finding is that good performance is linked to an effective combination of strategic control with attention to changing task demands, reflecting time and care taken in informing and effecting action. The contributions of this work towards our understanding of complex problem solving are the methodological approach to the creation of the model, the design patterns embedded in the model (which are a reflection of the cognitive demands imposed by the nature of the task) and mainly an explanation of how skill, described in terms of strategy use, is acquired in complex scenarios. This study also provides a deeper understanding of the interactions observed in the Cañas et al. (2005) dataset, including a computational realisation of how cognitive inflexibility occurs.

## Acknowledgements

I would like to recognize the instrumental role that my supervisor, Dr. Sharon Wood, played during the development of this work: her constant guidance and feedback, particularly during the writing of this thesis, made the completion of this work possible. I would also like to thank my thesis committee members, Dr. Peter Cheng and Dr. Richard Cox, for their valuable comments at the annual review meetings and the members of the *Representation and Cognition* lab for their friendship and willingness to help at all times.

I want to express my gratitude to the members of the *Cognitive Ergonomics* research group at the University of Granada (Spain), headed by Dr. José Cañas, for allowing me to use their data to construct the cognitive model. I also want to thank Scott Douglass and Dan Bothell for their help during the ACT-R summer school at CMU, where the construction of the model began and several design decisions were taken.

This work was founded by CONACYT (National Science and Technology Council - Mexico). My sincere gratitude for your support during all these years.

To God... for His precious presence, and His promises.

To my wife Jessica... for her unique ability to make me a better and happier person each day. Te amo.

To my son Daniel... for being an inexhaustible source of inspiration.

To my father… for reminding me of the value of being diligent.

To my mother... for encouraging me to focus attention on important things.

To my brother Alonso... for transmitting his enthusiasm to me and for our long chats.

To my brother Juan Carlos... for showing me what a good-hearted man is.

To my sister Kiki... for the strength she shows in everything she does.

To my friend Luis... for all the ideas we interchanged during this period.

# Table of Contents

## List of figures

## List of tables

# 1. Introduction

Given that solving problems is a pervasive aspect of everyone's life, it seems reasonable to dedicate effort to understand more about the nature of these problems and, ideally, to learn how to solve them. Researchers have created several categories of problems such as well-defined and ill-defined, static and dynamic, simple and complex problems, etc. (Newell & Simon, 1972). Considering these categories classic problem solving research has been focused on quite static problems such as the Tower of Hanoi (see section 2.3.1.1). In contrast, there are other tasks that are characterized by a salient dynamic component. The purpose of this research work is to conduct a detailed study of problem solving behaviour in a complex dynamic task, a fire-fighting computer simulation, in order to further understand strategy use in CPS. A methodology based on systematic data analysis of empirical data and mainly on a detailed modelling of problem solving behaviour is followed.

## 1.1. Problem solving, microworlds and cognitive modelling

Research in problem solving has provided useful constructs to explain how people solve problems such as the concept of problem space (Newell & Simon, 1972) which can be combined with the notion of purposeful behaviour to conceptualize a problem solver as an agent that produces a stream of behaviour when given a goal. This study is focused on a branch of problem solving named Complex Problem Solving (CPS). There is not a single definition of CPS available in literature. For example, Frensch & Funke (1995) consider CPS as an activity to overcome obstacles between current and goal states, whilst Anderson provides a quite operational definition: a "goal-directed sequence of cognitive operations" (1980, p. 257). For this research complexity is related the amount of variables involved in the problem solving situation, the number of decisions involved and perceptual-motor requirements. A paradigm to study CPS is taken from the work of Frensch & Funke (1995). In this paradigm a problem is not defined only by task features, but rather by "the interaction between task characteristics and person characteristics" (Frensch & Funke, 1995, p. 7).

Different aspects of CPS can be studied following this conceptualization of CPS. By focusing on the task only it is possible to explore different factors related to the structure of the problem: task characteristics (section 2.3.2) and task configuration (sections 2.2.3.3 and 2.2.3.4). Also, by isolating the problem solver, cognitive and non-cognitive variables (section 2.2.3.2) can be analysed. Furthermore, when the study considers the interplay between the problem solver and the task a new set of areas related to the process of coping with the task can be studied (section 2.3.6) including mechanisms such as the use of strategies, a topic largely studied in this research. As participants interact with the task there is a learning effect that has different components including skill acquisition (section 2.2.3.1), strategy consolidation (section 6.1.2.2) and cognitive inflexibility (section 3.1).

There are various approaches to the study of CPS. One of these approaches emphasizes the use of computerized simulations, also called microworlds. The work of Dörner (1996) is highly representative of this approach. Microworlds are computer simulations that represent a middle point between naturalistic scenarios and laboratory tasks (Brehmer & Dörner, 1993). Although microworlds are relatively simple, they embody the essential characteristics of real-world dynamic decision making environments (Gonzalez, Vanyukov & Martin, 2005). Microworlds have three characteristics (Gonzalez, Thomas & Vanyukov, 2004). The first characteristic is complexity, owing to the number of elements involved and the nature of their interrelationships (Frensch & Funke, 1995). The second one is the lack of transparency; the problem solver does not have access to all relevant task information, making interaction with the world necessary for knowledge requirements. And the third one is their dynamic nature: the problem state changes both independently and as a consequence of the participant's actions.

This research work is concerned with choice at the level of strategy selection, adaptation and implementation, and attempts to provide insights into interactions between these dimensions and performance under different conditions of task practice; therefore it is necessary to select a suitable dynamic task. This dynamic task is called FireChief (Omodei & Wearing, 1995). Imagine the following situation: you are in charge of fighting a fire spreading over a well-delimited landscape composed of different terrain types. In order to accomplish the goal of fighting the fire you are required to use two kinds of resources, copters and trucks, which have different capabilities. You also have knowledge about the direction and the strength of the wind. FireChief performance is measured in terms of the amount of terrain that is not destroyed at the end of a four-minute trial (a detailed description is provided in the next chapter). The specific kind of problem participants confront when dealing with FireChief is a dynamic decision problem (Brehmer, 1987) where a series of interdependent decisions are required to reach the goal, the environment changes over time, and the decisions change the state of the world (Brehmer, 1987; Brehmer & Allard, 1991; Gonzalez et al., 2004). In general terms, a FireChief participant is engaged in a strategic situation where he or she has control over a limited number of truck and copter fire-fighting units and has to use them to accomplish one mission: fighting the fire.

According to Schunn, McGregor & Saner (2006) a strategy is a coherent set of steps for solving a problem. For this research strategies are considered the basic construct to explain phenomena such as performance differences. When the problem solver is facing the situation imposed by the simulation, s/he must select actions that ultimately will lead to the achievement of a goal such as the successful control of the fire. It has been observed that everyone uses multiple strategies, and that different groups of people share many, if not most, strategies (Reder, 1987; Lemaire & Siegler, 1995). It has also been observed that participants vary in their distribution of use of strategies. These observations led to the creation of the strategy adaptivity approach (Schunn & Reder, 2001). In this approach while two individuals may have the same set of strategies, they may differ in their ability to select the best strategy for a given situation. Strategies are characterized by different levels of accuracy and required effort but, for Payne et al. (1993) an effortful decision process is identified by a high number of operators or more demanding operators. These topics of strategy use and implementation are explored in detail in this research.

Cognitive modelling has been used in dynamic environments such as air traffic control (Taatgen, 2005). As Frensch and Funke (1995) suggest, it is important to understand the process of complex problem solving rather than the product. For this research the problem solving product is comprised by a list of actions taken by the problem solver within a dynamic task. On the other hand, the problem solving process comprises a sequence of cognitive, perceptual and motor actions that should produce a sequence of FireChief commands. Insight into these cognitive processes can be gained through the construction of a cognitive model. How these actions are selected and executed by the model depends on the cognitive architecture that supports them.

A cognitive architecture embodies structures and mechanisms in the form of a general theory of how the mind works (Newell, 1990) and is used for creating simulation models of human cognition (Taatgen & Anderson, 2008). Cognitive architectures define a set of operations provided by their processing structure, in the form of mechanisms to access encoded knowledge (in the form of rules for example) to appropriately select actions to attain goals. One such architecture is ACT-R (Adaptive Control of Thought-Rational, Anderson et al., 2004). The basic principle of ACT-R is that an agent executes actions according to rational analysis: it selects actions that attempt to achieve its goals. According to rational analysis "each component of the cognitive system is optimised with respect to demands from the environment, given its computational limitations." (p. 29). A cognitive model developed in ACT-R makes use of the several modules of the architecture (for example, to perceive the environment or to remember things) and of its various mechanisms (for example, learning or action selection) to produce a stream of behaviour.

There are several cognitive modelling paradigms (Taatgen et al., 2006). In one of these approaches several strategies are implemented in the cognitive architecture and then compete for use in solving a problem. This paradigm is leveraged and extended in this research. According to Taatgen et al. (2006) utility learning is a useful mechanism in tasks where there are multiple cognitive strategies, but where it is unclear which one is best. The model is designed around the exploitation of the temporal utility learning mechanism embedded in ACT-R in order to provide the required sensibility considering the highly dynamic task it is facing. The characteristics of the task used in this research provided a challenging scenario for modelling. As a result, various experiences relating to the development of large cognitive models were documented throughout the various stages of the model's development. A cognitive model is implemented in the ACT-R cognitive architecture with the aim of gaining understanding about different aspects of problem solving described in the following section.

## The study of CPS behaviour

Now that the focus of this research is clearly defined, namely the modelling of strategy use in a complex dynamic task, specific issues can be addressed. A first challenge this research must tackle refers to the identification of strategies, that is, the ability to discriminate among distinct models of strategic behaviour based on empirical data. The kind of behaviour required by FireChief can be described as interactive in terms of Fu & Anderson (2008): "learning action sequences in situations that depend critically on the utilization of external cues" (p. 4). It is known that interactive behaviour in complex tasks is constrained by cognitive, perceptual and manual processes (Anderson et al., 2004; Taatgen, 2005). A cognitive model of a complex dynamic task can shed light on this topic by providing a detailed account of how behaviour is constrained by the various ACT-R modules controlled by the knowledge embedded in ACT-R. This research is focused on strategy use, including execution, adaptation and flexibility and is structured around specific research questions that are presented below.

### 1.2.1. What characterizes strategies in complex dynamic tasks?

The problem solver must achieve control over the situation by means of competent and time-constrained decision making. The complexity of FireChief is considerable: the landscape is composed of 400 blocks of terrain, there are 4 fire-fighting units, multiple fires spreading at the same time, three different commands available and the influence of wind strength and direction to be taken into consideration. The problem solver also needs to deal with the dynamic nature of the microworld. In this context two kinds of control seems to be in continuous competition (Taatgen, 2005). In top-down control the decision flow starts in the head of the problem solver, in his or her representation of the problem, and ultimately results in selection of an action that has some impact in the environment. This kind of control can be characterized as a plan that is followed. As actions are executed in the environment the state of the world is updated, and eventually it is possible to determine whether the goal has been achieved. In bottom-up control the problem solver executes actions based on the effects that they produce in the environment. The problem solver needs to be aware of these effects at all times; there is no pre-defined course of action but an adaptive selection of operators. This characteristic raises several questions such as which kind of control (top-down or bottom-up) should be exerted and when, or how to make sense of environmental feedback. Considering the speed at which events occur in FireChief psychomotor ability is a good candidate for good performance: the problem solver must issue a considerable quantity of commands under severe time constraints. For some researchers (Ackerman 1988; Ackerman & Cianciolo, 2002) task content is a determining factor of skilled performance. Because FireChief is spatio-temporal in nature (it requires the processing of spatial and temporal content), it is important to consider spatial ability. Spatial thinking requires the ability to encode, remember, transform and discriminate spatial stimuli.

### 1.2.2. How strategy use is affected by task manipulations?

The Experimental Psychology and Behaviour Physiology Department of the University of Granada kindly made available the data of 82 participants completing dynamic CPS trials using the FireChief simulation. This dataset was obtained using an experimental design that examined different aspects of strategy use, adaptation and strategy consolidation. The Cañas et al. (2005) study found interesting interactions using this data set. The results found by Cañas et al. are considered in this work but a brand new analysis was conducted. Because the ultimate goal was to create a cognitive model a new analysis at a finer level of detail than that carried out in the original study was required. Conducting this analysis also resulted in a detailed understanding of the FireChief task in which strategies are deployed. In the Cañas et al. (2005) study participants interacted with 24 FireChief scenarios. The first 16 scenarios were considered as the training phase and the last 8 scenarios as the testing phase. The various interactions found by the Cañas et al. (2005) study are re-examined here and more explanations are obtained through the new analysis and mainly by the explanations provided by the cognitive model. An important consideration here is how much of these explanations can be generalized to other complex dynamic tasks.

### 1.2.3. How do choices arise in complex and dynamic situations?

This study is focused on the interactive, dynamic decision making aspects of complex tasks. Fu & Anderson (2006) consider that the ability to make quick, non-deliberative decisions that occurs through the exposure to the same or similar situations is a major component of performing complex skills. Because decisions in this kind of task must be made under considerable time-pressure, the problem solver needs to select appropriate actions quickly. Any benefit derived from making a decision decreases with the amount of time it takes to be executed (in the extreme case, the best decision will become completely useless). Adaptation makes sense when environmental conditions are not always the same (Schunn et al., 2001), and adaptivity is directly related to sensitivity to environmental change. The problem solver uses feedback for determining the effectiveness of his or her interventions. Gonzalez et al. (2004) affirmed that performance in dynamic tasks is highly determined by the ability of the problem solver to recognize that it is necessary to alter the decision processes. Problem solvers require continuous processing of feedback in order to select appropriate actions within an ever-changing situation (Brehmer & Dörner, 1993). The cognitive model prescribes a mechanism in which environmental feedback controls how actions are selected in a highly dynamic task.

## 1.3. A challenging task

One of the main difficulties associated to the use of microworlds for experimental research is how to deal with their inherent complexity. A data analysis tool was developed to aid in the data analysis process. This same tool is used to assess the Quality of Fit of the model with the empirical data. This tool was used to analyse and make sense of empirical data in order to identify and characterize participants' strategies.

An important aspect that needs to be considered when developing an ACT-R cognitive model is that the task must be available to the model. This is a technical requirement that is imposed on the modeller. If the task is simple enough this requirement may represent a dozen LISP functions. In the case of FireChief this requirement represented a rather challenging technical endeavour: a new version of FireChief was developed for this research in the LISP language. A particularly difficult aspect of this task was to handle all the possible events that could arise in FireChief while the model was executing commands and perceiving events.

Cooper et al. (1996) stress the gap that exists between an informal psychological statement and the computer realisation of this statement in a computer language such a LISP using the ACT-R cognitive architecture. This research work shows how this gap is bridged by defining a cognitive modelling approach. The cognitive modelling literature provides a set of principles that were used during the development of the model, for example Taatgen (2005) advocates the most simple control structure. The cognitive modelling paradigm followed in this research shares several elements with the *Strategy Behaviour Paradigm* described in Taatgen et al. (2006). Nevertheless, there are many characteristics of the FireChief task, mainly its dynamic component, which requires a new approach to its modelling. For example, there is a considerable amount of perceptual and motor processes that must be executed at a fast pace, so it is necessary to find a modelling approach that addresses these requirements.

Another topic related to the modelling of complex tasks is how we can exploit the information given by the model. An ACT-R cognitive model provides a trace of all the actions that are being executed in the different modules and the content of all its memories, this information is essential for understanding the underlying process, but there is another potential source of information that is usually not considered by modellers. In a cognitive architecture such as ACT-R there is a formidable amount of computation happening at all times used during the conflict resolution phase to select a single action for the next cycle. An approach that uses the utility of productions to understand more about the underlying phenomenon is proposed and used in this research.

As Anderson et al. (2004) put it: the external world provides much of the cognitive tissue that integrates cognition. One of the motivations for this study is to assess the capabilities of ACT-R for dealing with a complex problem such as FireChief. The authors of FireChief postulate that this microworld captures many of the properties of real world fire-fighting (Omodei & Wearing 1995). The idea of investigating cognition in a complex task was compelling. In ACT-R coordination of behaviour depends on the central production system (which stores and uses production rules), but it is aware of a limited amount of information: just that which can be stored in registers that can hold only one piece of information at a time (Anderson et. al, 2004). The FireChief microworld requires a considerable amount of processing by the visual and motor modules and its procedural memory is composed of several hundreds of rules.

## 1.4. Overview

The approach taken in this research is as follows. First, relevant knowledge about microworld characteristics is gathered from the literature. This literature survey is centred on complex, dynamic and non-transparent microworlds and the demands they impose on problems solvers. Second, a particular microworld, FireChief, is selected and a task analysis is conducted. This task analysis is detailed enough to allow a re-implementation of this task in a different programming language. Third, a large set of data is analysed and several behavioural patterns are extracted and organized into strategic categories, significant interactions are documented also. This analysis is assisted by the use of a software tool developed for this end. Fourth, a fine-grained cognitive model is developed with the aim of replicating the main interactions observed in the Cañas et al. (2005) study. The model should also provide a detailed explanation of these phenomena, including the considerable variability shown by participants.

This research is interested in understanding more about how people deal with complex dynamic situations, particularly at the level of strategy use, so the different choices made by the problem solvers are the focus of attention. In the FireChief model there are different kinds of decisions. First, a strategy must be chosen. In the FireChief model a strategy is composed of a set of *intentions*. So the second kind of choices refers to the selection of the next intention to be executed. After an intention is selected there is a third level of choices: how to execute the intention. An intention is executed by selecting FireChief commands. After a FireChief command is selected the model decides how to execute this command through a combination of cognitive, perceptual and motor actions. The ACT-R cognitive architecture allows the modelling of these different kinds of decisions.

## 1.5. Plan of the thesis

Chapter 2 reviews theory related to Complex Problem Solving (CPS). The first part discusses topics related to CPS such as its definition, determinants of CPS performance and different approaches to its study. This chapter also introduces a particular set of tasks called microworlds. Much of the literature discussed in this chapter is centred on the use of computer simulations for the study of CPS behaviour. Concrete examples of microworlds are provided with the aim of illustrating the concepts described in the chapter. In this section a comprehensive description of the FireChief microworld (Omodei & Wearing, 1995) is provided. The chapter continues with a discussion of the various cognitive demands associated with the use of microworlds before describing the advantages and difficulties associated with the use of these simulations. Because the methodological approach followed in this research is cognitive modelling, chapter 2 ends with a section related to cognitive architectures and cognitive modelling. A description of the ACT-R cognitive architecture is included in this chapter alongside the reasons behind its selection in this research.

A focal point of this research work is how to overcome the complexity inherent to the use of microworlds for psychological research. Chapter 3 describes an approach based on task and data analysis. Chapter 3 starts describing in detail the Cañas et al. (2005) study, which is the source of the dataset used for cognitive modelling. To support the data analysis phase a software tool called the Protocol Analysis Tool (PAT) was designed and developed. The way this tool enables the analysis of FireChief data is discussed in this chapter. The main outcome of the analysis of the data is the definition of a set of strategies. These strategies are named, described and organized into a hierarchy. The second half of chapter 3 presents a series of statistics and metrics extracted from the data. Significant interactions discovered through this detailed analysis of the data are presented here. This section also presents a special set of metrics related to the use of FireChief commands which are particularly relevant in assessing the Quality of Fit (QOF) of the model.

The potential benefit of any cognitive architecture is the opportunity to bring to bear multiple constraints in a single framework. The sources of constraints that impact the model’s behaviour are depicted in the CPS paradigm proposed by Frensch & Funke (1995): the problem solver, the task and the environment. Previous chapters describe different aspects of the FireChief task and the cognitive demands it poses to problem solvers. Chapter 4 is about how the complex problem solving process happens from the problem solver’s perspective. The ACT-R cognitive architecture is an abstraction of the problem solver’s cognitive system, and the FireChief task is solved by making use of the various mechanisms built into the architecture. This chapter describes with more detail these mechanisms. The model’s behaviour is also determined by its knowledge; Chapter 4 therefore describes which knowledge is available to the model. This chapter explains that the general approach followed during the development of the model was to enforce the free competition of small blocks of behaviour based on environmental rewards.

Bearing in mind the research goal of increasing our understanding of strategy use in dynamic tasks, chapter 5 describes the results obtained with the model. This chapter starts with a description of the output of each model run and the analysis of the Quality of Fit (QOF) of the model using several measures. Interesting findings related to command use are presented in this section too. A particularly relevant section of this chapter describes how the procedural knowledge embedded in the model, governed by a set of utility values, was tuned to the task by the continuous interaction of the model with the simulation and, among other topics, how the different training programmes mediated this process. Knowledge about how to execute actions is given to the cognitive model at the outset, but the exact emergence of strategies is a product of the interaction between the problem solver (the model) and the environment. The last chapter, chapter 6, presents a general discussion of the contributions of the thesis which are organized around the explanations of CPS behaviour and performance provided by the cognitive model. This chapter also presents the conclusions of this research work.

# 2. Complex Problem Solving, Microworlds and Cognitive Architectures

This chapter provides the context in which this research is conducted. The first two sections introduce concepts related to problem solving and Complex Problem Solving (CPS) respectively including a theoretical framework for the study of CPS. The third section describes computer simulations called microworlds and explains why they are suitable for studying CPS. This section also discusses the demands imposed by this kind of simulation bearing in mind the ultimate goal of laying the groundwork for the development of a cognitive model. Given that human rational behaviour is constrained by the structure of the task and computational capabilities of the problem solver (Simon, 1990) it is possible that the various demands that complex dynamic tasks impose on problem solvers, such as the fact that a high level of dynamics reduces the available time for making decisions dramatically, may exceed their cognitive abilities. This chapter ends with a description of cognitive architectures and cognitive modelling.

## 2.1. Problem Solving

Solving problems is a pervasive aspect of everyone's life. From the many definitions of problem solving (cf. Frensch & Funke, 1995) it is possible to extract a set of commonalties. First, problem solving is goal-directed (Scheiter et al., 2000). This means that it is necessary to find the sequence of actions that will produce the desired outcome using whatever knowledge and techniques problem solvers have. By generating these actions the environment is affected and therefore the problem solver needs to evaluate the consequences of his or her actions and act accordingly. Second, problems are decomposable. For instance, the Tower of Hanoi (TOH) problem of moving several disks from one peg to another can be decomposed into sub-problems related to moving single disks to different locations. Nevertheless goal decomposition requires the execution of processes for coordinating the execution and completion of sub-goals, placing extra cognitive demands on the problem solver. Third, the problem solver relies on abstractions. Some problems can be represented as problem spaces (Newell & Simon, 1972) comprised by problem states and operators. A key assumption of this approach is that a problem space should only include aspects of the task environment that are relevant for solving a particular problem. This means that the problem solver creates a useful representation of the task. This abstraction enables a searching process driven by two elements: which operators apply in the current situation and the probability for each of being on the correct path. Fourth, problems solvers make use of methods, or strategies. The available strategies are determined by the chosen representation. Methods are organizations of behaviour directed towards solving problems. There is a huge variety of strategies that can be leveraged for solving a problem. Some of them are simple: the answer to a problem may be evoked from memory or a problem may be solved by a "*generate and test*" approach. Others are a bit more elaborate such as means-ends analysis. The nature of certain tasks requires the application of very specific strategies which can hardly be transferred to other scenarios. Because each strategy has a different probability of solving the problem at hand it is important to understand how strategies are selected and implemented. Fifth, problems can be classified. A common distinction is between well-defined problems, where the problem solver is provided with all the information needed in order to solve the problem; and ill-defined problems, where the problem solver needs to make an extra effort to define the problem. Another one is between simple and complex problems, a topic discussed in the next section.

## 2.2. Complex Problem Solving

In traditional problem solving, where research emphasized the process of moving between intermediate states until the final solution is reached, there was a preference for using tasks where these intermediate states are physical such as the Tower Of Hanoi (VanLehn, 1989). These tasks are novel to participants, have clearly defined optimal solutions, are solvable within a short time frame, and participants' problem solving steps can be traced. In this respect Funke (1991) argued that it is hard to generalize results obtained from traditional problem solving studies to Complex Problem Solving (CPS) due to the low validity of simple laboratory tasks with respect to the complexity of real-life problems. Also Quesada, Kintsch & Gomez (2002), who were interested in defining CPS, consider that due to the unclear definition of CPS it is hard to say whether traditional problem solving findings can be extended to CPS. This kind of discrepancy spawned different approaches to the study of CPS. In one of them, the American approach, there was an interest in studying problem solving in domains that require extensive knowledge, such as reading, writing, arithmetic, social and natural sciences, and games (Sternberg & Frensch, 1991).

An alternative approach, branded the European tradition, presents two variants: the first related to the work of Broadbent (Broadbent 1993; Berry & Broadbent, 1988) and the second represented by the work of Dörner (Dörner & Wearing, 1995). The Broadbent tradition emphasizes the distinction between cognitive processes that operate under awareness versus outside of awareness. Berry & Broadbent (1995) used a task environment called Sugar Production where an input variable (workforce) is manipulated in order to control the output of the system (sugar production) at a pre-specified level. It was observed that although people are capable of controlling the system (i.e. maintaining the targeted amount of sugar production) they are unable to verbalize the relation between the variables. On the other hand Dörner's tradition emphasizes the use of computerized simulations. For example in the SINUS simulation there are three input variables that control three output variables and the relation among them is governed by three linear equations, and the task that the problem solver confronts is to understand the nature of these equations. Within this same tradition there are also microworlds that highlight the dynamic aspects of problem solving such as fire-fighting scenarios (Brehmer 1987, Dörner & Wearing 1995, Omodei & Wearing 1995, Rigas et al., 2002), emergency dispatching (Joslyn & Hunt, 1998), and air traffic control (Taatgen, 2005).

### 2.2.1. An operational definition of CPS

The definition of CPS followed in this research was labelled the "Gap definition of CPS" by Frensch & Funke (2005). This gap is a separation between the current state and the desired state, and problem solving is conceptualized as a process of overcoming obstacles to achieve goals. The problem solver can overcome these obstacles by means of behavioural and/or cognitive, multi-step activities. In traditional problem solving the presence of a single barrier is common while in CPS there are multiple co-existent barriers. Consequently, in CPS a long series of decisions is required to solve the overall problem and each new decision influences the conditions for the next one (Quesada, Kintsch & Gomez, 2002).

### 2.2.2. CPS paradigm

The Gap definition of CPS structures a paradigm of CPS focused on the interaction between task, environment and problem solver (figure 2.1). This interaction creates a CPS situation where the problem solver applies resources (cognitive and non-cognitive) to overcome obstacles. In this view the solution to the CPS situation implies an efficient interaction between the problem solver and the task environment (Wenke & Frensch, 2003). The importance of studying the problem solver and the task at the same time is also stressed by Brehmer & Dörner (1993). This paradigm also dictates that any transition from a given state to a goal state is constrained by the problem solver's memory content, information processing capabilities, and by the tools that are available to the problem solver (Frensch & Funke, 1995).

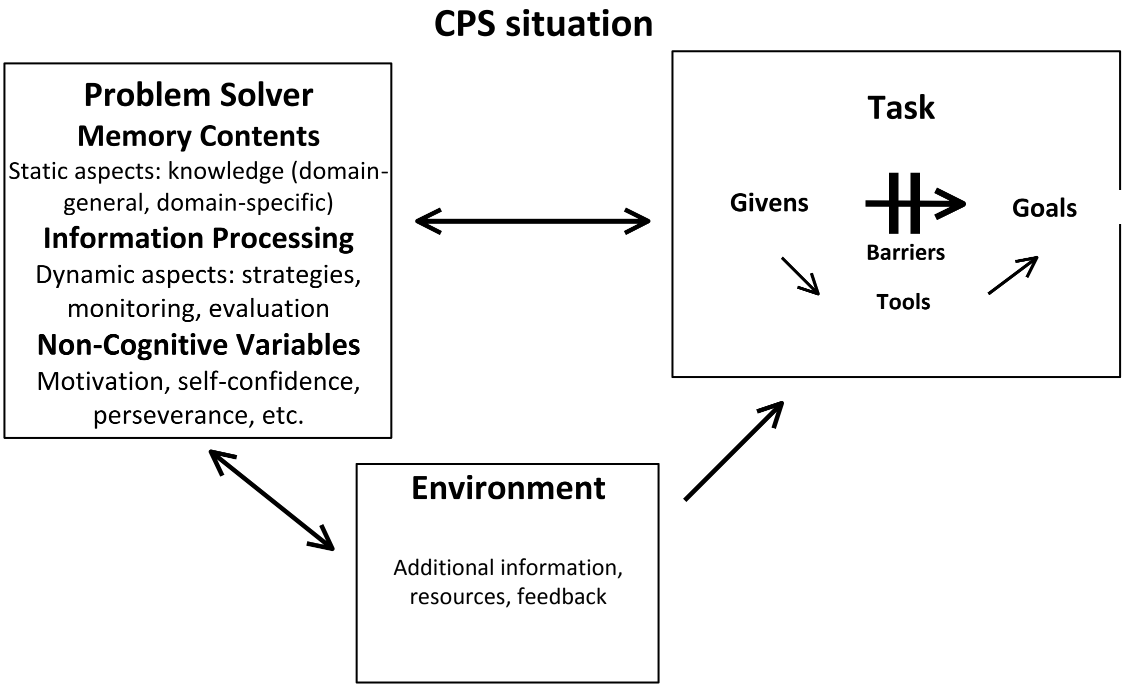



**Figure 2.1: CPS situation (taken from Frensch & Funke, 1995, p. 22)**

### 2.2.3. Determinants of CPS performance

From the problem solver's point of view, several determinants have been identified as playing a role in CPS performance.

#### 2.2.3.1. Experience

A skill acquisition theory has been tested using Air Traffic Control (ATC) tasks (Ackerman, 1988, 1992; Taatgen, 2001). According to Ackerman (1988) there are successive stages during skill acquisition. Initially, during the cognitive stage, behaviour is error prone and participants need to be more focussed on the task instructions. In the associative stage, participants memorize some of the rules for landing planes (i.e. instructions) and error rate is reduced. Finally, in the automatic stage, participants have dominion over the rules of the task, speed increases at the same time that errors are reduced. Experience also affects the strategies that are employed by the problem solver. It was proposed by Anderson & Lebiere (1998) that, as experience in a task is gained, the participant increments his or her performance by proceduralizing knowledge. Another explanation is that better performance is achieved when there is a more efficient schedule of perceptual, cognitive and motor processes. Two factors mediate changes in ability-performance relations due to practice: complexity, which implies greater reliance on general intellectual abilities, and consistency (uniformity of information processing demands across practice).

#### 2.2.3.2. Cognitive and Non-cognitive factors

Cognitive variables represent different abilities the problem solver can make use of for solving the task, such as information processing capacity, working memory and psychomotor ability. By analysing the information processing requirements of a task it is possible to determine a candidate set of these factors (Ackerman, 1992). Moreover researchers have used individual measures of these factors to predict performance in various tasks (Gonzalez et al., 2004; Rehling et al., 2004). For Dörner (1996), failure is not random but exhibits common patterns and the objective of his research is to discover how to improve our ability to control complex systems and reduce errorful behaviour by analysing their roots. Dörner identified various sources of errors, of faulty behaviour, when people deal with complex tasks, for instance insufficient goal elaboration, neglecting side and long-term effects (by not considering the interaction of variables and insufficient coordination of problem solving actions). When considering the use of strategies, Dörner identified two sources of errors: failure to detect inappropriate strategies, and failure to repair inappropriate strategies. One of the computer simulations used by Dörner to study these kinds of errors is Moro, a game involving planning the survival of a semi-nomadic tribe in the African Sahel (see section 2.3.1.4). Dörner also stresses the importance of self-reflexive examination in order to adapt to the given circumstances. In considering non-cognitive factors, Dörner identified a tendency to guard one's feeling of competence as a reason behind the lack of adaptation. By not looking at one's mistakes one's estimation of one's own competence is not endangered. In other example, Hesse et al. (1983), using a computer simulation called Epidemic, found that highly distressed participants obtained higher quality values, worked harder, took more effective actions, and recognized effective measures more readily. Brehmer & Dörner (1993) argue that emotions are elicited by success or failure, so participants need to cope with their emotions when dealing with this kind of computer simulation.

Substantial differences have been observed among individuals interacting with CPS situations (Taatgen, 2001; Schunn & Reder, 1998; Rehling et al., 2004) and different approaches to study these differences have been developed. One of these approaches is the *parameters approach* which assumes that differences in performance can be attributed to fundamental differences in the cognitive system. Two examples or parameters are processing speed and working memory capacity (Taatgen, 2001). According to Gonzalez et al. (2003) very little is known about which cognitive abilities are necessary for successful performance on dynamic tasks (the relevance of this kind of task is explained later) although Ackerman (1988) suggests that spatio-temporal task domains such as computer simulations have an important spatial component and research has found positive correlations between working memory and perceptual/motor ability with performance in an Air Traffic Control (ATC) task (Rehling et al., 2004) and between working memory, proceduralization speed (ability to learn new rules) and psychomotor speed (time needed for a key-press) with performance in another ATC task (Taatgen, 2001). Various studies have found that processing/reasoning ability and learning potential tend to be related to good performance (Funke & Frensch, 1995). A parameter can be defined at a higher level, for example as "the ability to solve problems". In this respect Funke and Frensch, based on the evidence of the importance of context, semantic embeddedness and knowledge, argue that there is no abstract context-independent capacity for solving problems. On the other hand Joslyn & Hunt (1998) developed a task for measuring the abstract decision making ability of participants. They found that performance in this abstract task was positively correlated with performance in an ATC and an emergency dispatching task.

#### 2.2.3.3. Environmental factors

Environmental factors affect CPS by constraining the information processes that can be applied, by influencing information accessibility and by restricting which means can be used for solving the problem (Frensch & Funke, 1995). Environmental factors include the nature of feedback, expectations about the task and the possibility or necessity of cooperation with other problem solvers. Using a simulation called Temperature, in which participants control the temperature of an artificial system, Heineken et al. (1992) used an experimental design with three conditions: no delay, short delay and long delay when providing feedback concerning the quality of the interventions, that is, if the manipulation made to the system served or not to achieve the desired temperature. They found that feedback delay reduced performance. Another study looking at the impact of feedback delays in a fire fighting scenario was conducted by Brehmer (1995). The delays were due to slow reporting by the fire-fighting units about their location and actions. These delays force participants to coordinate their decisions by remembering information needed to organize their sub-goals and thus place demands on working memory. As a consequence of these delays not every action shows immediate consequences so participants need to have a way of representing the future state of the system. Expectations about a problem solving task are usually provided in the form of instructions. In the contexts of computer simulations these instructions range from a very precise set of rules to ambiguous statements such as: "improve the welfare of a tribe". Group problem solving has also captured the attention of several researchers (Johansson, Persson, Granlund & Mattson, 2003; Gonzalez et al., 2004); in this case the problem solver needs to communicate and perhaps cooperate with other problem solvers.

#### 2.2.3.4. Task manipulations

Various studies have found that specific task manipulations have an impact on performance. Task manipulation is paramount within the context of CPS due to the very nature of the experiments, where the degree of control over the development of the experiment is low. Alibali & Booth (2002) studied the influence of cues on the formation of representations. The materials used in this research were ten problems about constant change; for example, one of these problems is to calculate the total growth of a sapling taking into account its growth rate. They manipulated the wording of the problem and the use of graphs to vary the intervention. Their hypothesis was that some cues can enhance the representation of the problem and hence improve strategy selection. They found that graphical cues play an important role in enabling strategy discovery by providing the means for problem solvers to create mental models of the task. Del Missier & Fum (2002) studied the toads and frogs puzzle (refer to section 2.3.1.2 for a description). The study was focused on the role of hints to improve performance. They defined a set of strategies and studied the way participants select among them. They manipulated the availability of hints: for one group the interface provides information about the movements that were legal. A legal move is one that complies with all the rules of the game. They also implemented a cognitive model (cognitive models are discussed in section 2.5.1) for each of the four main strategies. The dependent variables used for assessing the quality of fit of the model with the empirical data is the number of trials needed to interchange all the toads for all the frogs in two consecutive trials (the researchers consider that if the task can be solved twice in a row it can be considered as mastered by the problem solver) and the error percentage. After comparing the results of the four cognitive models with the experimental data they determined that participants shift strategies as experience grows and fewer memory demands are placed upon them. Charman & Howes (2002) studied the effect of highly constrained goal definitions on strategy generation. For one experimental group the task was to design two layouts: one for a computer room and one for a study area; the other experimental group was asked to copy a layout. Both groups use a software package for creating the layouts. In this study it was found that participants with a strong goal definition (design a layout) executed more mouse commands than participants in the not-strong goal group, indicating the use of inefficient strategies that require a larger number of mouse commands. A more efficient strategy used the mouse to select multiple items and then apply an action to all these items, therefore reducing the number of commands issued in comparison to an inefficient strategy where a single item is selected before applying the desired action. The authors argue that the strong goal task reduced the opportunity for the use of the most efficient strategies that involve working with multiple items at the same time.

Problem context, also called semantic embeddedness, is also a determinant of CPS. The way the problem is presented to the person affects their chance of success, and the likelihood of transferring this success to a new problem (Kotovsky, Hayes & Simon, 1985). In an experimental study (Bechmann & Guthke, 1995) use a microworld similar to SINUS (Funke, 1991). In this microworld there are three inputs that affect the value of three outputs and this effect is controlled by three linear equations. In the first stage of their experimental design participants interacted with the microworld for 21 simulation cycles. According to the authors, during this stage participants acquire knowledge about the structure of the task. In a second stage a target state (specifying values for the three output variables) is given to participants and they must reach and maintain this state for seven simulation cycles. Semantic embedding was manipulated by offering to participants two different cover stories: in the first one a “cherry-tree description” was used. In this condition the input variables were light, water supply and warmth and the output variables were cherries, leaves and beetles. In the second condition an abstract machine description was used, so variables had no meaning. In both conditions the microworld is governed by the same equations. The authors hypothesized that the abstract version of the task would not activate prior knowledge. In order to test their hypothesis, after each simulation cycle participants were asked about the existence of causal relationships between variables and a measure of knowledge acquisition was obtained for each participant. Participants' performance in this microworld was measured as the Euclidean distance of participants' interventions and the optimal intervention (considering the desired target) and the result was averaged over the seven simulation cycles. This measure was considered by the authors as a measure of knowledge application. They found a correlation between traditional psychometric tests and performance only in the abstract condition. The authors concluded that "the semantic embeddedness of a complex problem mediates the relation between traditional measures of intelligence and CPS performance." (p. 194). This finding can be transferred to other microworlds where there is considerable semantic embeddedness such as FireChief. The authors also found that knowledge acquisition only occurred in the abstract condition. In this respect Beckmann and Guthke (1995) argue that the familiarity caused by the semantic context may have led the participants to develop "sham confidence" that interfered with knowledge acquisition. This study highlights the importance of problem context in CPS: the amount of learning may be mediated by the semantic content of the task.

Veksler, Gray & Schoelles (2007) studied how information access influences decision making. They used the *Table Decision Task* that displays a table "…in which each of six alternatives (arranged in rows) had a value on each of six attributes (arrayed in columns). The value of the alternative was derived by summing the attribute scores so that the higher the value, the better the alternative" (p. 1). The goal of the task is to select the best choice based on the relative values of its attributes, so the most effective strategy is to check the value of all the cells. In an experimental condition there was an extra cost in time associated with disclosing the information in each of the cells. This task illustrates how system characteristics can be altered for testing hypotheses. The authors of the study predicted that performance would vary based on exploration/exploitation costs that variations in the task environment imposed on the decision maker. This task was used to study how information access influences cognitive and perceptual-motor trade-offs. According to Veksler, Gray & Schoelles (2007) comparison of durations between commands can provide insights into how strategies evolve. Fu & Gray (2006) used a Map Navigation task where participants were given a start station and destination (end station), and were asked to travel from the start station to the end station. The researchers manipulated two variables: the cost and utility of information. They argued that the activity of "checking transfer time" represented the critical explicit information-seeking action and designed an experiment to determine if people are able to adapt their information-seeking actions based on the cost of performing them. The way they manipulated this cost was similar to the study of Veksler, Gray & Schoelles (2007) discussed previously: they added a lockout time after the transfer station was clicked in order to gather transfer time information. Utility of information was manipulated by controlling the difference between the speeds of the slow and fast transfers (larger differences represented higher utility). They found that people adapt to the cost of accessing information by spending more time seeking transfer time information even when the cost of this activity is high in terms of time.

Chung & Byrne (2004) studied a failure called the post-completion error (omissions of actions required after the completion of a task's main goal). An example of a post-completion error given by the authors is "forgetting to remove the original after making a photocopy" (p.1). They tried to reduce its occurrence by creating appropriate visual cues. They controlled the availability of cues for helping participants remember to execute a final complementary task. The study identified false completion signals as the source of this kind of error and proposed that visually distinguishable cues presented at the right time can reduce its occurrence. Many real world environments can benefit from understanding why cognitive errors such as these arise (such as aerospace navigation).

#### 2.2.3.5. Strategy use

As noted by Brehmer & Dörner (1993) when conducting research with complex tasks it is hard, if not impossible, to duplicate the same set-up for every participant (refer to section 2.3.5) so the study of complex behaviour should be focused on strategies and tactics. According to Schunn, McGregor & Saner (2006) a strategy is a coherent set of steps for solving a problem. Strategies can be classified in two groups: weak and strong (Jonassen, 2000). Weak strategies include hill climbing (small changes are applied to a random solution, each time improving it a little) and means-ends analysis (Anderson, 2000), weak strategies do not take advantage of the domain characteristics. Strong strategies are domain-specific. When people interact with microworlds for the first time, it is to be expected that they will try to apply weak problem-solving procedures to the knowledge that they have about the domain in question (Anderson, 1983). For Anderson, weak methods can be fed with knowledge acquired from the domain so that it is possible to apply them successfully. Another way of classifying strategies is to distinguish between compensatory and non-compensatory strategies (Kerstholt, 1992, Payne et al., 1993). Compensatory strategies involve the consideration of all relevant information that the environment can provide, whereas decision-makers using non-compensatory strategies generally utilize only a subset of the available information. For example, in Kerstholt's study, a choice task involving selecting between apartments was used. Task complexity was manipulated by varying the number of known attributes, such as rent and noise level: either 4 or 8. During the task, a number of attributes were queried and one of them was selected. It was found that task complexity reduced the depth of search in this task; in other words, non-compensatory strategies appeared as a response to task complexity. In a typical non-compensatory strategy a low value for the most important attribute implies that an option is discarded, regardless of the value of the other attributes. In general, compensatory processes are thought to lead to better choices, but at a cognitive cost (Payne et al., 1993). Given the levels of complexity and dynamics of the FireChief task, it is anticipated that the type of strategies used by participants in this microworld will be non-compensatory.

It has been observed that there are many different ways in which a task can be solved and individuals vary in the strategies they use (Schunn & Reder, 1998). Strategy selection can be explained by a combination of differences in knowledge (i.e. the participant either knows or does not know a strategy), cognitive factors (i.e. the participant is able to carry on the necessary mental processes for executing a strategy), and non-cognitive factors (e.g. psychomotor ability is necessary for executing the commands). It has also been observed that different groups of people share many, if not most, strategies (Reder, 1987; Lemaire & Siegler,

1995) and that participants vary in their distribution of use of strategies. These observations led to the creation of the strategy adaptivity approach to individual differences in CPS performance (Schunn & Reder, 2001). In this approach while two individuals may have the same set of strategies, they may differ in their ability to select the best strategy for a given situation. This approach assumes that people vary in their general ability to detect situational change and/or select strategies appropriate to the new situation (Schunn & Reder, 2001).

From a methodological perspective during an initial phase it must be proven that individuals are in fact using different strategies and after that it is necessary to determine how individuals are choosing these strategies, for instance that participants are able to adapt their strategy choice and execution to the changing environment. This feature becomes particularly relevant considering the ever changing characteristics of microworlds. Schunn & Reder used an ATC task for exploring strategy use; they focus mainly on proving that it is possible to predict performance in this ATC task by defining and measuring strategy adaptivity (see section 2.2.3.6). Wenke, Frensch and Funke (2004) also highlight the importance of studying differences in knowledge and strategies. In this view behaviour is determined by knowledge and architecture (cognitive), and interactions between these areas explain differences in performance. This observation is important as the strategies that participants deploy in computer simulations are usually context dependent, mostly because they are executed to solve tasks that require the use of domain knowledge.

Strategy choice is also constrained by the characteristics of the task. A study by Lovett and Anderson (1996) using the Building Sticks Task (where participants have to construct a stick of a certain length using an unlimited supply of sticks of smaller lengths) found it was possible to create a bias for either the *undershoot* or *overshoot* strategies by manipulating the length of the target stick. On the other hand, rational approaches assume that participants choose strategies according to their costs and benefits (Anderson et al., 2004). Neth & Payne (2001) studied the influence of different interface configurations for the solution of simple addition problems presented on a computer screen. In this task participants need to solve 72 addition problems presented in lists of 4, 8 or 12 single-digit numbers. Interface properties such as the colour, the location of numbers and the possibility of moving the numbers in the list were manipulated. In one experimental condition it was not possible to interact with the numbers forcing participants to rely solely on their memory; another condition allowed some degree of interaction such as the manipulation of location of numbers. The authors hypothesized that ordering the numbers on the screen facilitates the task by grouping numbers that add up to a round number. The study concluded that participants are sensible to the cost of accessing information, and that this sensitivity produces significant strategy differences. When more interactive features were available to participants, such as the option of moving the numbers in the list with the mouse, participants adopted a strategy that exploits this new feature. Schoelles and Gray (2001) used an Air Traffic Control task that requires participants to assess whether the aircraft in the radar are hostile or not. To make this evaluation participants use the display to query the attributes of every aircraft and apply a set of rules. One of the experimental groups had a visual aid which marks which objects have already been evaluated and it was observed that this aid improved their performance by avoiding re-selection of aircraft. All these results reinforce the idea that subtle changes in the interface may correspond to significant cognitive changes.

### 2.2.3.6. Strategy adaptivity

Although some participants use the same set of strategies they can differ in their ability to opportunistically apply strategies in response to the situation (Schunn & Reder, 2001). Gonzalez et al. (2004) affirm that performance in dynamic tasks is highly determined by the ability of the problem solver to recognize that it is necessary to alter their decision processes. Adaptation makes sense when environmental conditions are not always the same (Schunn et al., 2001), and adaptivity is directly related to sensitivity to environmental change. Research has found that the problem solver keeps track of the base rate of success for each strategy, that is, the proportion of times a strategy is successful for his or her problem (Schunn et al., 2001). As task characteristics change, individuals experience different success base rates for each strategy and learn to prefer different strategies (Schunn et al., 2001). In this respect the authors give the example of an adaptive participant in an arithmetic task: they choose to *calculate* in early trials, *retrieve* in the final/later trials, and to use a mix of strategies in the in-between trials. Strategy selection is also related to recognizing which problems are familiar. In the case of arithmetic problems, the problem solver can decide to retrieve when presented with very familiar problems. These two kinds of adaptivity can be branded extrinsic and intrinsic respectively. Extrinsic adaptivity uses information external to a particular problem (i.e. learning the base rates of success) while intrinsic adaptivity uses information internal to a particular problem (i.e. familiarity with the problem).

According to Schunn & Reder (2001) extrinsic strategy adaptivity can be measured at two levels: micro and global. At the micro-level there is sensitivity to the successfulness of a strategy on the immediately preceding attempt at using that strategy, similar to simple operant conditioning (Skinner, 1953). In their study using the KA-ATC task (section 2.3.1.7) they were focused on a strategy for landing aircraft. Sensitivity was detected at the micro-level because participants use this strategy more often when it is successful in the previous attempt. They focused their analysis on the first four trials because the rate of error decreases as experience is gained in the task. Sensitivity at the global level refers to sensitivity to changes in the frequency of success defined over an accumulation of past strategy attempts. This kind of sensitivity is closely related to the concept of utility in ACT-R (section 2.4.1.4) where the history of success is a component of its total value. Schunn & Reder conclude that strategy adaptivity at both levels is an important component of performance in complex, dynamic tasks. In the study of Schunn, Lovett & Reder (2001) using the Building Sticks Task (see section 3.4) it was found that explicit awareness of success rate, which was measured by applying a questionnaire at the end of the experiment, is related to adaptivity and is positively related to performance. Also So & Sonenberg (2004) consider situation awareness to be essential in complex dynamic task environments and they stress the importance of switching between top-down and bottom-up modes of control when dealing with these kind of tasks. This observation is particularly relevant for development of the cognitive model and will be discussed in chapters 4, 5 and 6. Friman & Brehmer (1999) found that building situation awareness is essential when dealing with battle situations and that a commander needs to develop sensitivity to changes in the battlefield. Friman and Brehmer consider that the critical dimension in a dynamic process is time, and that the problem solver must make correct use of information regarding time; that is, the problem solver needs to be aware of the different time scales between decisions, actions and results. Time scales result from time constants, for example, in FireChief this time constant refers to the duration of commands. Situation awareness is directly related to bottom-up control because it depends on opportune perceptions of the environment but is also supported by the representation of the task.

Some researchers use a similar concept to strategy adaptivity: cognitive flexibility (Cañas et al., 2005). For Jonassen (2000) cognitive flexibility is considered to be a form of cognitive control. It is thought that problem solvers with higher levels of cognitive flexibility will outperform less flexible ones due to the fact that the former tend to consider more alternatives (Stewin & Anderson, 1974). But exploring more alternatives is not the only characteristic of flexible problem solvers; according to Kerstholt (1996) another condition for adaptive strategy selection is that the effect of a strategy on both accuracy and effort is *known* by the problem solver. Studies of strategy adaptation range from the retrieve vs. compute paradigm (Lebiere et al., 1998), and the Building Sticks task (Lovett & Anderson, 1996), to Air Traffic Control (Schunn & Reder, 2001). Schunn and Reder (1998) were able to measure strategy adaptivity by detecting decisions points for strategy selection and using problem solvers' responses at these points. Using the KA-ATC task (section 2.3.1.7) the authors focused their attention on the stage at which a runway for landing the aircraft is selected. At this decision point participants can either select a long or a short runway. They observed that there are many advantages related to use of the long runway (there are fewer restrictions for using it and it involves fewer keystrokes) but that the use of the short runway allows the long runway to be kept free for aircraft that must land on it. They conclude that in order to obtain good performance participants must maximize the use of both runways and defined an adaptivity measure called *optshort. Optshort* is calculated as the proportion of times a participant opts to land an aircraft on the short runway relative to the total number of the times an aircraft is landed when both runways are open. They manipulated the number of 747s, which must land in the long runway, presented in each trial to increase/decrease the opportunity to make the strategic decision *optshort* and found that this measure correlates with general performance ($r=.69$, $p<.0001$). They also found that a different mix of aircraft (that is, different task complexity) used in a trial produces a strategy shift for more adaptive participants. The authors of the study found a positive relation between how much people adapt to the task and two psychometric measures: WM (measured using the Four-Term Ordering test) and inductive reasoning (using the Figure Sets, Figure Series and Figure Matrices tests). A description of these psychometric tests is provided in Schunn & Reder (2001).

## 2.3. Microworlds as complex problem solving task domains

Brehmer & Dörner (1993) commented on the difference in the amount of complexity between field research and laboratory studies: in the former there is too much and in the latter there is too little. They also present three reasons that explain why psychology is now better able to handle complexity. First, information technologies facilitate the analysis of larger sets of data. Data can be stored in well-structured databases and can be queried at will for obtaining a rich set of metrics. The second reason is the availability of means for the modelling of very complex models of human functioning. For example cognitive architectures, described later in this chapter, provide a means for modelling human behaviour. And the third reason is the existence of tools for executing complex experiments in the form of microworlds, which is the topic of this section.

In order to complete the picture provided by the CPS paradigm of figure 2.1 it is necessary to describe a microworld. A microworld is a realistic environment that allows the study of various kinds of responses to complex and realistic decision making situations (Omodei & Wearing, 1995). It presents to the participant a number of different problems, rather than a single, well-defined task (Brehmer & Dörner, 1993). Microworlds require the continuous application of controlled information processing, imitating in this way naturally occurring problem solving environments. It has been observed that the underlying structure of the task must be congruent with participants' pre-existing understanding of the task (Omodei & Wearing, 1995), for example in a fire-fighting microworld simulated fire should behave in a way that reflects the dynamics of real fire. Microworlds bring the complexity of real-life problems to the lab in a way that can be partly controlled (Quesada, Kintsch & Gomez, 2002). Although microworlds can be relatively simple, they embody the essential characteristics of real-world dynamic decision making environments (Gonzalez et al., 2004) and can be considered as a middle point between naturalistic scenarios and laboratory tasks (Brehmer & Dörner, 1993).

According to Omodei & Wearing (1995) there are three kinds of microworld. First there are computerized training simulators. An example of this kind is the Air Traffic Control (ATC) simulator. This kind of microworld can be used either for skill acquisition or for training in process control. In this kind of task it is important to understand and apply rules correctly. For example in the Kanfer-Ackerman ATC task (Ackerman & Kanfer, 1993) participants need to land different kinds of aircraft on two kinds of runway. After selecting a plan for landing, participants need to apply the rules for landing aircraft correctly in order to gain points. Another task of this kind is emergency dispatching, where participants need to classify incidents reported by a text interface and execute specific actions; in this case participants also need to apply a specific set of rules designed to attend to incidents properly.

The second type of simulation is comprised by recreational video games. A difficulty posed when dealing with video games is that they are usually too complex to use as task environments for conducting research and it is not possible to modify the parameters of the game for experimental purposes. Omodei and Wearing (1995) argue that it is usually hard to adapt this kind of microworld for precise experimental studies, although there are some interesting attempts in this respect such as the Open Real-Time Strategy project (ORTS) (Buro, 2004). ORTS provides a rich and fully customisable environment for running scenarios in which participants can take control of buildings and units and can play against other participants or AI agents. The complete source code is available making it possible to develop any number of scenarios, units, actions and data logging functions. Laird (2002) used simulated virtual environments to make it possible to study real-time decision making and interaction while reducing the considerable difficulties posed by sensing and acting in the real world. The third kind of simulation or microworld is specifically developed for research into complex decision making processes. The European approach represented by Dörner has used a large quantity of these simulations, some of which are described in the following section.

### 2.3.1. Eight problem solving tasks

In order to provide a clear picture of what a microworld is eight problem solving tasks domains are described in this section, including a comprehensive description of the FireChief task due to its relevance for this research work.

#### 2.3.1.1. Tower of Hanoi (TOH)

TOH is a well-defined task. There are three pegs and a fixed number of disks of different sizes (see figure 2.2). In the initial state all disks are stacked in descending order on a single peg. The objective of the game is to move all disks to the central peg. TOH is also knowledge-lean because there are only two rules: only one disk may be moved at a time (a move consists in moving the upper disk of a peg onto a different peg) and no disk may be placed on top of a smaller disk. The optimal solution to a problem with 3 pegs and 3 disks requires 8 movements and there is an algorithm for solving a problem of 3 pegs with 7 discs (although with more pegs finding the optimal solution is not that straightforward). TOH is time-invariant: the only way of changing the state of the task is by applying the "Move" operator, for this same reason there in no time pressure. It is a planning task because there is no advantage in reacting to the environment. Performance is measured as the number of moves required for moving all disks to the target peg.

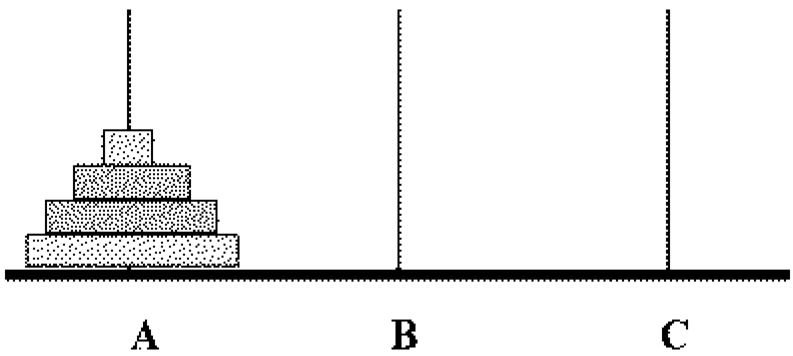



**Figure 2.2: The Tower of Hanoi task**

#### 2.3.1.2. Frogs & Toads (F&T)

In the Toads & Frogs puzzle (Berlekamp, Conway & Guy, 2001) "... three toads are placed on the three leftmost squares of a seven-square board while three frogs are placed on the three rightmost squares. The central square is initially empty. The goal of the game is to switch the animals' positions by having the toads occupy the three rightmost, and the frogs the three leftmost squares, respectively. A square can be occupied by only one animal at a time, and an animal can move only into an empty square. Toads can move only rightward and frogs only leftward. There are two possible types of move: a Slide to the next (empty) square and a Jump over an animal of a different type to a two-square apart empty position. A solution can be reached in exactly 15 moves, 9 jumps and 6 slides" (Del Missier & Fum, 2002, p. 1). Figure 2.3 shows the user interface of this task, note that there are seven cells in total and only one is empty at a time. As in the case of the TOH this task is well-structured. It is knowledge-lean because it suffices to know a couple of rules and how to move the pieces. It is time-invariant and there is no time pressure. It is a planning task. This task has an optimal solution (cf. Del Missier & Fum, 2002). When solving this task the problem solver needs to decide between 'slide' or 'jump' (although sometimes his or her decisions are forced). Performance is measured as the number of moves required for switching the frogs with the toads.

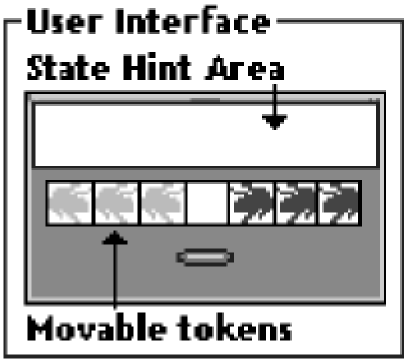



**Figure 2.3: The Frogs & Toads task (taken from Del Missier & Fum, 2002)**

#### 2.3.1.3. Coldstore

In the Coldstore microworld (Reichert & Dörner, 1988) participants need to manually control the temperature of a warehouse. Figure 2.4 shows the interface of Coldstore. The goal is to keep the temperature stable at an optimal temperature in order to keep products from freezing or spoiling (top indicator in figure 2.4). In Coldstore each trial lasts about 12 minutes. Performance on this task is measured by the total number of deviations from the goal temperature. This task is time-variant. There is a single input variable: to increment or decrement temperature (participants use the control wheel displayed at the bottom of figure 2.4) but the effect of modifying this variable has a delay before taking effect, and the temperature varies continuously with time. The numbers of the control wheel do not correspond to the cold store temperature. Participants can change the position of the control wheel by clicking the arrows with the mouse (large arrows change temperature more). Variables in Coldstore are highly interconnected: the relation between the thermostat and the resulting temperature is governed by a linear equation. This task is also non-transparent because the linear equation is not known and there is a feedback delay. Güss et al. (2004) report that participants have the impression that the temperature does not immediately react to changes on the control wheel. This task is reactive because participants need to act in response to variations in temperature. It is well-defined because the goal is clear. According to Rigas et al. (2002) the best strategy in this microworld is to "wait and see"; excessive interventions are not recommended.

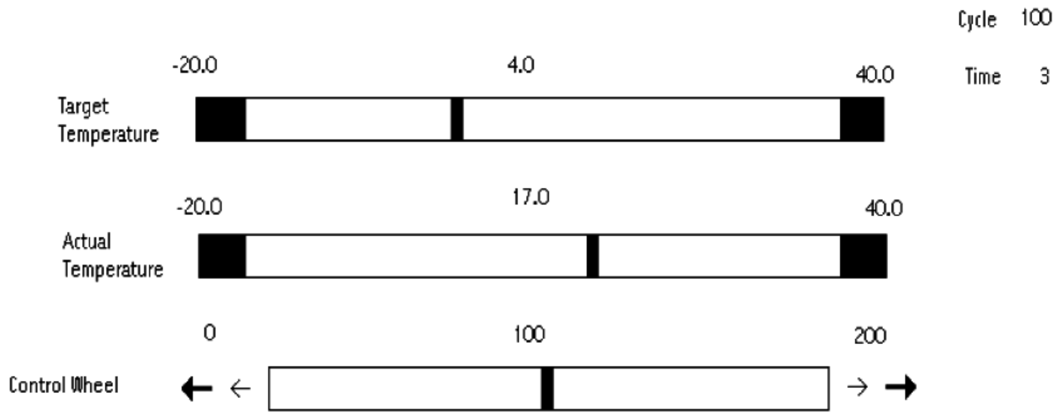



**Figure 2.4: The Coldstore task (taken from Rigas et al. 2002)**

### 2.3.1.4. Moro

In the Moro microworld (Dörner, Stäudel & Strohschneider, 1986) participants need to take a series of decisions in order to improve the quality of life (the goal given to the problem solver is not clearly defined) of the fictitious Moro tribe. These decisions are related to various issues such as the control of population, health measures and the development of the economy. For Dörner (1996) complexity is strongly related to the number (and variety) of variables and Moro has many. Figure 2.5 shows some of the most relevant variables in Moro and how they influence each other. For instance, an increment in *Pasture Land* reduces *Erosion (of the land)*, while an increment in *Cattle* causes more *Erosion*. Variables are highly interconnected: there is a complex network of interdependencies not disclosed to participants and some variables are hidden, therefore this microworld is non-transparent. Moro is time-variant; it changes on its own, without intervention. Nevertheless, unlike FireChief, there is no time pressure in this microworld, as Moro is an event-driven simulation: it changes in response to inputs given by the participant. The interface of Moro is not presented here.

When interacting with Moro participants are aided by a person who knows how to input the different variables. This microworld is knowledge-intensive: there are several concepts that participants need to understand in order to make good decisions. It is also a planning-problem; Dörner (1996) identified the different stages that the problem solver follows when interacting with Moro: gathering information, making decisions for particular problems of the tribe, and executing these decisions. The problem solver receives feedback from his or her actions and is able to gather further data, and make new decisions.

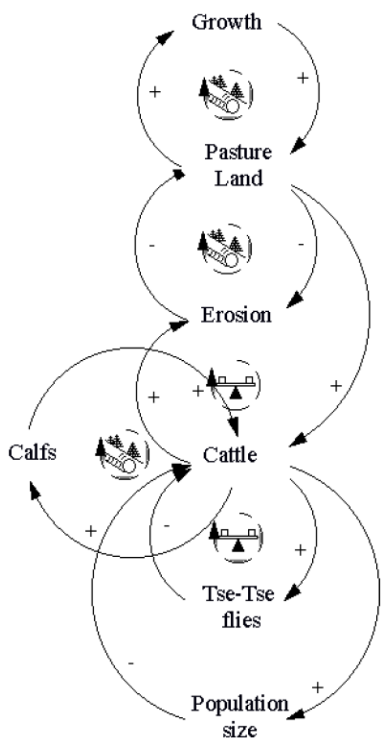



**Figure 2.5: The relationship between variables in Moro**

### 2.3.1.5. FireChief

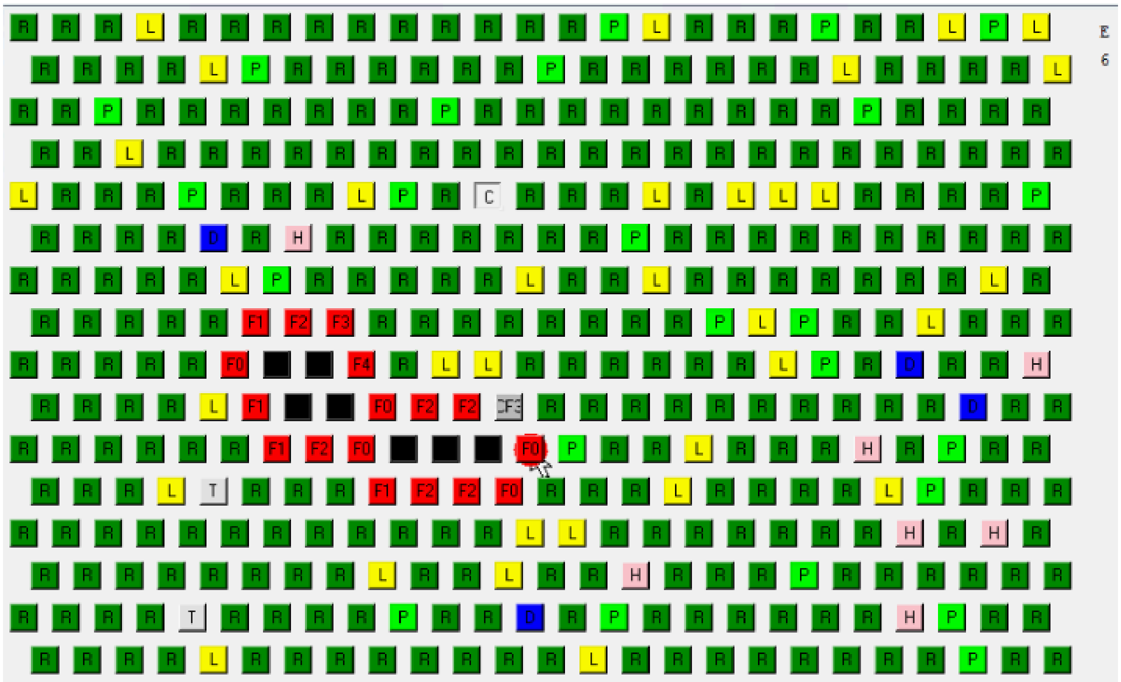


**Figure 2.6: the FireChief task (LISP version)**

In the FireChief task (Omodei & Wearing, 1995) players engage in complex high-level decision making activity involving the use of fire-fighting units combating a spreading fire with the goal of preserving as much landscape as possible for a predefined amount of time. A participant's overall performance is measured as the percentage of unburned landscape at the end of the simulation. At the end of the trial the participant is informed of the performance for that trial. In this task several events can occur when the simulation state is refreshed every 200 milliseconds. Figure 2.6 shows the new version of FireChief created in LISP to enable interaction with ACT-R. This new version follows the same structure and is governed by the same equations that the original one. The task environment is comprised by a matrix of 25 columns and 16 rows. Each element in this matrix represents a cell of terrain. There are different types of terrains with different values (considered when computing final performance). The values of these elements are: forest (4, colour green), pasture (6, colour light green), clearing (2, colour yellow) and house (30, colour pink). Problem solvers are informed about these values. When a cell is on fire its colour turns to red. A special type of landscape is the dam (in blue) where appliances can refill tanks; a dam can never catch fire. Terrain elements are labelled, this label is used to identify detailed information such as the intensity of the fire (for instance the fire located at the upper left has an intensity of 1). Blacks cells are burned out.

### A. Units

In FireChief there are two types of units: copters (grey) and trucks (light-grey). Copters are three time faster than trucks and, because they are airborne, cannot be destroyed by the fire. Trucks are slower than copters but their tanks have twice the capacity (10 units). The main difference between trucks and copters is that the former are able to execute *Control Fire* commands. The trucks also have a value and because these units can be engulfed by the fire it is necessary to protect them as well. FireChief uses different icons to distinguish between copters and trucks, when units occupy a specific cell the icon of that unit fills that particular cell, in this way participants can know at all times the location of units. In the LISP version (see figure 2.6) cells occupied by trucks have a different shade of grey than the ones occupied by copters and a cell containing a truck is labelled “T” whilst a cell containing a copter is labelled “C”.

### B. Commands

Execution of FireChief commands is accomplished by three mechanisms: moving the mouse pointer, clicking the mouse and pressing keys. Actions from the mouse (move pointer and click) can be executed on every terrain cell. Only two keys can be pressed: a “D” for *Drop Water* and a “C” for *Control Fire*. Commands are executed by fire-fighting units. At any one time, fire-fighting units may be inactive, in motion, extinguishing a fire, controlling a fire, or refilling its tank at a dam.

The empirical data collected in FireChief is mainly related to command execution. The first row in table 2.1 shows the fields collected for each command, a collection of these entries comprises a FireChief protocol which is the empirical data collected for each participant. The first column keeps track of the command number. The second column is the command type, the third column represents the time in which the command was executed (in seconds). The fourth column is the current performance. The fifth and sixth columns represent the unit number and unit type respectively. The seventh column shows the Cartesian location of the command for the DW and CF commands and in the case of a *Move* command the seventh column is the starting point and the eighth column the ending point of the movement. The last column shows the type of terrain in which the command is executed (for the *Move* command this is the terrain type of the destination location). Figures 2.7 (a) to (j) are snapshots of a run executed by the cognitive model; they illustrate how FireChief commands are executed. The resulting protocol is also presented to show how empirical data is collected.

| Number | Type | Time | Performance | Unit # | Unit Type | origin | destination | terrain |
|---|---|---|---|---|---|---|---|---|
| 7 | Move | 20.978 | 98.2 | 3 | Copter | 9,11 | 11,11 | Forest |

***Table 2.1: FireChief protocol the command described in figures 2.A to 2.D***

### Move

Copters and trucks can be moved to any cell in the landscape matrix. In order to execute a Move command the mouse cursor must be located over a unit and the mouse button must be pressed. After that the mouse pointer must be dragged to the target cell and the mouse button must be released. FireChief disables the unit for an amount of time directly proportional to the Euclidian distance between the cells (the type of fire-fighting units is also considered, copters are faster than trucks). When the Move command is completed the unit has moved to the new location and is ready to execute another command. When a unit is moved to a dam the tank is automatically refilled after 4 seconds.

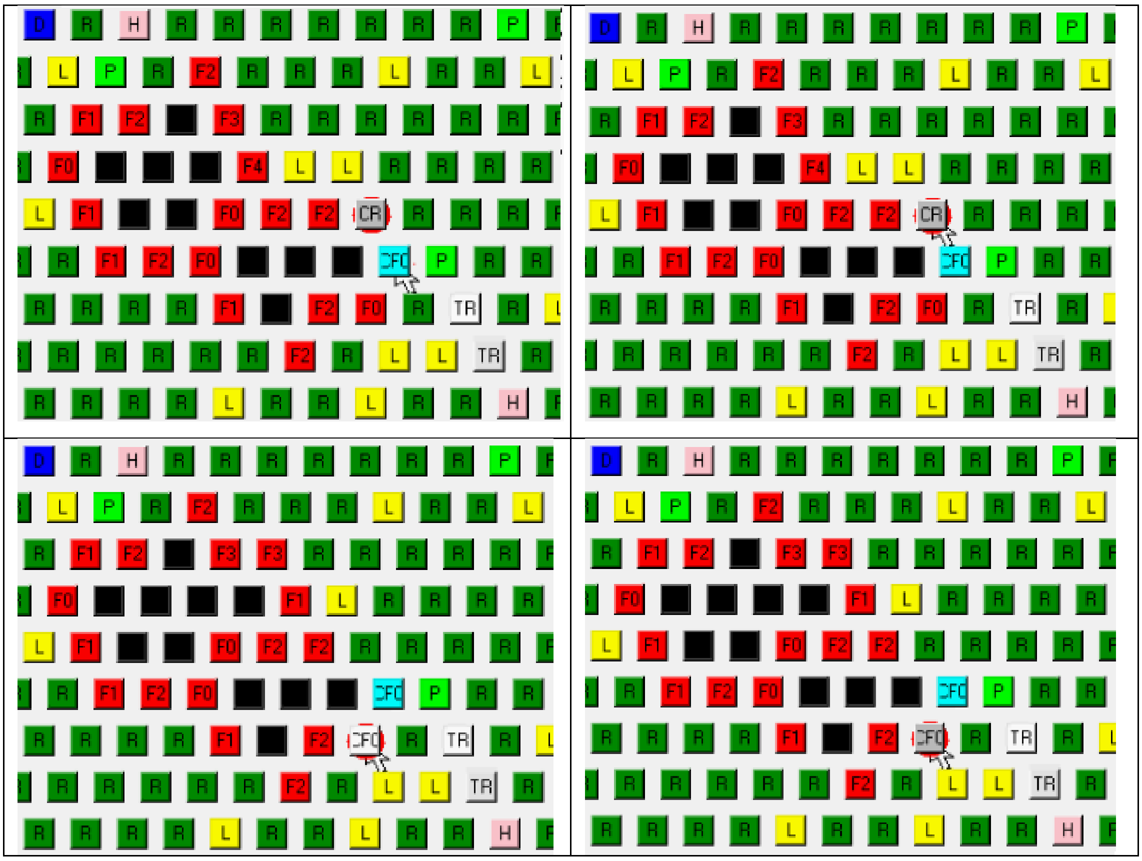


**Figure 2.7 (a) (upper-left) A copter (CR in grey) is idle. 2.7 (b) (upper-ri7ght) The mouse pointer is located over the idle copter. 2.7 (c) (bottom-left) The copter is dragged two cells below, the copter starts moving and is disabled (white colour). The fire destroyed a cell. 2.7 (d) (bottom-right) the copter arrives and is idle again.**

In figure 2.7 (a) the mouse cursor is over a cell where a copter is executing a DW command over a fire of intensity 0 (CFO). Attention is placed over a cell of type Forest where a copter is available (CR). The other copter is executing a DW command ("CFO", in light-blue). Several fires are destroying cells to the left of these copters. Trucks are located to the right of the copters. One of the trucks is idle (the one in light-grey) and one is moving (the one in white). In figure 2.7 (b) the mouse is moved over the previously attended Copter and the mouse is clicked. No other changes in the environment are visible but the fire is destroying the inflammable material of several cells. In figure 2.7 (c) the mouse is placed on a different cell (two cells below) and the mouse's button is released. This action completes a *Move* command and therefore the copter is disabled for the time required for a two-cell movement.

Table 2.1 shows the resulting entry in the FireChief protocol. Notice that six commands were executed previously: four *Move* commands to move all the units and two *DW* commands. Performance is not 100 because three cells have been destroyed: [8,8] (second 10.708), [7,7] (second 18.964) and [8,9] (second 20.975). In figure 2.6 these cells are not destroyed. It is possible to see that cell [8,9] with label F4 is destroyed in the transition of figure 2.7 (b) to (c) The destruction of this cell produced two fires: one of intensity F1 to the right (this fire is of intensity 1 because it is a clearing) and one of intensity F3 to the upper-right (a Forest). In Figure 2.7 (d) the Copter arrives at its destination (showing a grey colour).

There is an important functional dependence between the *Move* command and the other two FireChief commands. That is, before executing either a *Control Fire* or a *Drop Water* command, a fire-fighting unit must be located in the appropriate coordinate. In this sense the execution of a *Move* command can be considered as a pre-requisite to execute the other two commands. This functional dependence is discussed in further chapters.

## Drop Water

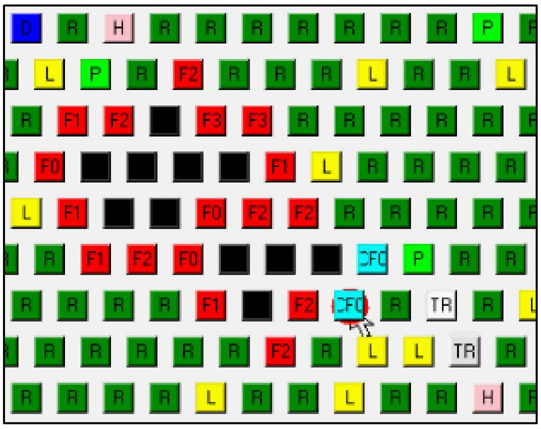



***Figure 2.7 (e) The copter that was moved to a new location executes a DW command***

Units must have water in their tank before performing a *Drop Water* (DW) command. The DW command cannot be cancelled so the problem solver must wait until its completion. To execute a DW command the unit must be placed in the desired spot and the mouse pointer must be over the unit. The Drop Water starts when the "D" key is pressed. If the command can be executed (there is water in the tank and the fire strength does not exceed the fire-fighting capability of the unit) the unit will start dropping water on the spot for 4 seconds (during this time the unit is disabled). If the DW is successfully completed the fire will be extinguished but the damage done by the fire is still there (each cell has a limited amount of destroyable material, if this amount is equal to 0 the cell is considered destroyed). This same cell can catch fire again in the future.

In Figure 2.7 (e) the "D" key is pressed while having the mouse pointer over the cell where the Copter is located starting a DW command. The cell has now a light-blue colour. It is possible to see in table 2.2 that the DW command is executed after the arrival of the Copter. The origin cell is equal to the destination field of the previous command.

| Number | Type | Time | Perf. | Unit # | Unit Type | Origin | Destination | Terrain |
|---|---|---|---|---|---|---|---|---|
| 8 | Drop Water | 21.937 | 98.2 | 3 | Copter | 11,11 | | Forest |

***Table 2.2: FireChief protocol for the command described in figure 2.E***

## Control Fire

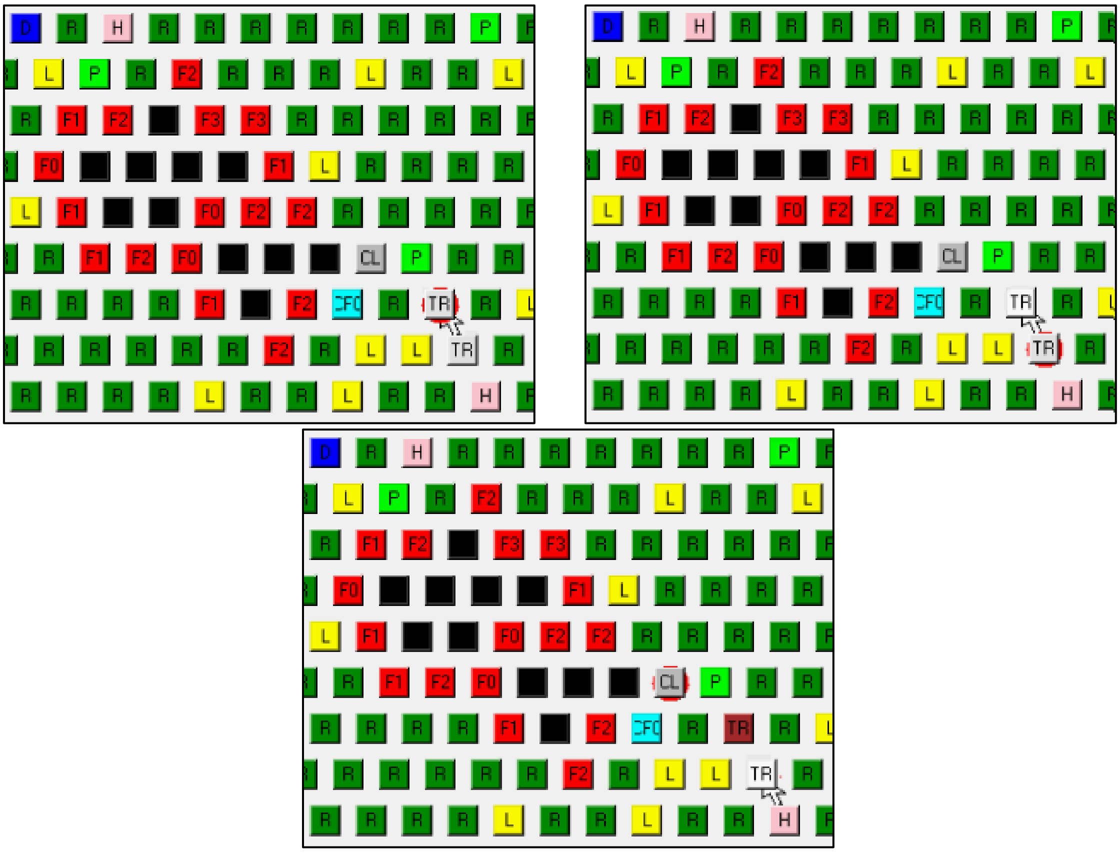


**Figure 2.7 (f) (upper-left) the mouse is moved over the idle truck (light grey). One of the Copters finishes a DW command. Figure 2.7 (g) (upper-right) The CF command is executed. Figure 2.7 (h) the CF is completed.**

To execute a *Control Fire* (CF) command the mouse pointer must be located over the unit and the "C" key must be pressed. If the command can be executed (the cell is not on fire) the unit starts removing the inflammable material on that spot and finishes after 2 seconds (the unit is disabled during this time). There are two conditions under which the command cannot be completed: the command can either be cancelled by moving the unit to another cell or by a fire developing in the same spot. If the command is successfully completed the cell changes its colour to brown and the fire cannot destroy that cell and therefore cannot cross that cell to reach adjacent cells.

In Figure 2.7 (f) the mouse is moved over the idle Truck (light grey). Notice that one of the DW commands is now completed (the Copter is now in a grey cell) but the DW command executed in figure 2.7 (e) continues its execution. In Figure 2.7 (g) the “C” keyboard is pressed while having the mouse pointer over the cell where the Truck is located starting a CF command. The cell has now a white colour (for the next two seconds). In Figure 2.7 (h) the cell where the CF was executed is now in in brown indicating that the CF is completed. The other truck is executing a command and is in white.

| Number | Type | Time | Performance | Unit # | Unit Type | origin | destination | terrain |
|---|---|---|---|---|---|---|---|---|
| 9 | Control Fire | 23.289 | 98.2 | 1 | Truck | 11,13 | | Forest |

***Table 2.3: FireChief protocol for the command described in figures 2.6 (f) to (h)***

Table 2.3 shows that the CF is executed right after the DW described in the previous section. The current performance is the same because no more cells were destroyed.

### C. Fire Development and wind conditions

If a cell is on fire, on each cycle (of .2 seconds) FireChief subtracts an amount of fuel according to the type of landscape, the intensity of the fire and the wind strength. If the total amount of fuel is depleted the cell is considered destroyed and fire propagates to the adjacent cells. If a cell is destroyed fire spreads to its six neighbouring cells with varying intensities. Fires are more intense in a forest than in a pasture, a fire in a clearing has low intensity, and fire spreads quicker in a pasture than in a forest. Although landscape distribution influences the 'shape' of a fire, wind strength and direction are the most determining factors because fires burn most intensely in the direction towards which the wind is blowing. Another important phenomenon is a change in wind strength, because fires can increase or decrease their strength depending on the size of the variation. This change of size happens suddenly modifying dramatically the complexity of the task. Not all fires change their intensity after a switch in wind strength; the ones that are affected are those located in the direction of the wind. So for example, if there is a circle-shaped fire and the wind is blowing to the right, if the wind increases its intensity only fires located to the right of the fire will increase their intensity. Wind strength ranges from two to eight and can blow towards the right, up-right, up, up-left and left. Wind strength and direction are displayed in the upper right-hand corner of the display (in figure 2.7 (f) the fire is blowing to the right with an intensity of 6).

### D. Alarms

Alarms are emitted due to various situations: when a unit’s tank is depleted and a *DW* command is attempted, when a unit is executing a *DW* command and the problem solver tries to interrupt it, when the problem solver tries to extinguish a very strong fire (that is beyond the appliance's capacity to extinguish it) or when a truck is in danger of being destroyed by the fire. Alarms are presented in the form of sounds.

FireChief is non-transparent because many aspects of the simulation, particularly those related to fire behaviour, are not visible to the problem solver. FireChief is non-linear; the equations that describe transformations to the fires based on wind intensity, flammability, humidity, etc. are not linear, there is a strong interconnection of variables. The intensity of emerging fires (also called spot fires) depends on the current wind strength; if it is above 7 the spot fire will have an initial intensity of 4. Timely identification of emerging fires is important as they can represent a threat due to an increase in wind strength. Following the task specification of the Cañas et al. (2005) study, for all trials in this research if the participant (or the model) gains control over the fire, at most two spot fires will appear at random locations and the participant will have to deal with them.

#### 2.3.1.6. Terminal Radar Approach Control (TRACON)

In the Terminal Radar Approach Control (TRACON) microworld participants interact with a display that shows several aircraft and the task is to acknowledge and monitor these aircraft as they approach different sections of the display. Aircraft are indicated by cursors, with a label next to them indicating flight designator, altitude, bearing, and speed information. Information about commands and procedures is available to participants. Trial complexity is manipulated by adjusting weather conditions and the number of aircraft (Ackerman, 1992). Another way of controlling the complexity of this microworld is by manipulating the kind of flights that appear in each scenario. There are three kinds of flight that can be handled: over-flights, arrivals and departures. According to the study of Ackerman (1992) participants make about six times more manoeuvre commands for arrivals than for over-flights. It was found that trials with a higher number of arrivals imposed higher information-processing demands on participants. In the same study another dimension of complexity was used: some aircraft did not show full compliance to participants' commands. The author proposes that there should be a clear distinction between complexity due to changes in content (arrivals vs. over-flights) and the compliance condition (not a change in content). Cañas et al. (2005) use a similar experimental design where the efficiency of fire-fighting appliances is reduced for one test group (this is similar to the compliance condition in TRACON) and the nature and behaviour of fires is manipulated in another (similar to changes in content in TRACON). Both kinds of changes affect performance in the aforementioned tasks by increasing complexity.

#### 2.3.1.7. Kanfer Ackerman Air Traffic Control Task (KA-ATC)

KA-ATC (Ackerman & Kanfer, 1993) simulates various aspects of real life air traffic control. The objective of the task is to land as many aircraft as possible in the available time (the task consists of a sequence of 10 minutes trials). There are twelve hold positions for the aircrafts divided into three levels, and four runways for landing the aircraft. Important variables that need to be considered in this task are the weather (that changes approximately every 30 seconds) and the wind direction and speed (because they affect the condition of the runways). Messages are given when there are changes in the weather and when the participants make errors. There is a set of rules that the participant must follow to land the aircraft such as that

747’s may never land on the short runway. This task can be decomposed into three unit tasks: moving an aircraft between the hold-levels, landing an aircraft on a runway and getting an aircraft from the queue into a hold-position. Landing the aircraft involves finding an aircraft to land, selecting the aircraft, moving the aircraft, finding a runway, moving to the desired runway and landing the aircraft (Ackerman, 1988). Each one of these actions requires a sequence of keystrokes, shifts of attention and encoding of information to be executed. In other words, issuing commands in microworlds, such as ATC task environments, requires a combination of cognitive, perceptual and motor actions and each of these actions demands resources in order to be completed. In this example, the subtask of moving an aircraft between the hold-levels requires a set of cognitive and perceptual actions that ultimately produce a sequence of motor commands; however the specific way of combining these actions is directly related to the strategy that is followed.

#### 2.3.1.8. Emergency Dispatching

In the Emergency Dispatcher task (Franklin & Hunt, 1993) a computer-controlled display shows information about emergency incidents and available resources. The dispatcher can receive messages from a call receiver and from field units. Each new message is announced by a beep and then appears at the bottom of the dialogue window. Only three messages are available at a time in the display; earlier messages can be retrieved again but with a time penalty. The dispatcher needs to confirm the classification of incidents and assign the nearest available and appropriate unit to respond. It is also possible to use a “surprise element”. For example, consider an incident that at first appears to be a routine traffic stop. In this situation the dispatcher may send an officer to check why the motorist has stopped his or her car, but after five minutes the motorist draws a weapon on the officer. The goal in the Dispatcher task is to evaluate the available information and to decide which emergency incident to deal with next and which unit to dispatch; participants were instructed to attend to higher priority incidents first. In general terms participants are responsible for assigning personnel to emergency incidents and monitoring the status of each incident until it is resolved. In the study of Joslyn & Hunt (1998) complexity in the Dispatcher task is determined by the number of incidents presented over a 15-minute period. Every incident has a code (there are 15 different kinds of incident) and there is a priority level for each of them. In this study, 54 participants (aged between 18 and 22) interacted with this microworld for 3 sessions of 1 hour each. During the initial training session, participants handled 9 incidents using 8 units. A second period of practice involved 15 incidents with 10 units. Finally, during the testing session, there was a mixture of the same complexity as in the previous sessions with an increase of 2 or 5 incidents.

### 2.3.2. Characteristics of microworlds

The gap definition of CPS discussed in Section 2.2.2 introduced the concept of obstacles that the problem solver must overcome; this section explores the nature of these obstacles in the context of microworlds. Researchers in this area (Gonzalez et al., 2004, 2005; Brehmer & Dörner, 1993) have identified a set of relevant dimensions for characterizing microworlds which are discussed in the following sections.

#### 2.3.2.1. Complexity

Complexity is related to the number of elements present in the system and the number (and nature) of the relations among these elements, therefore complexity is a function of the interplay of variables (Funke, 1991). It has also been observed that problem solving quality decreases as a function of the increasing number of variables as well as the number of non-linear functions (Hussy & Graznow, 1987). About this last point Dörner (1996) found that people tend to underestimate exponential growth. Complexity in microworlds is also related to the number of possible actions that a participant can take in a single moment. In these tasks there is often more information available than a human being can process at one time (information overload); this implies that the problem solver is responsible for the allocation of their own limited cognitive resources. Attneave (1954) proposed that given that our capacity to process separate items of information is limited, much of what is presented in the stimulus is ignored. This fact favours the exploitation of points of maximum information. Microworlds usually present the problem solver with an environment rich in visual elements. In this situation the problem solver needs to process a subset of this visual information whilst being aware of the consequences of his or her actions. This rich environment can be used as an External Memory (EM). For tasks such as microworlds the EM is comprised by the immediately available visual field and the opportunity for making use of this EM is available. In one study Fu & Gray (2001) found that the problem solver calculates the cost of accessing this EM and then decides either to use EM or internal memory.

#### 2.3.2.2. Transparency

Transparency refers to the availability of information about the problem. Funke (1992, 1995) hypothesized that a participant exploring a dynamic task environment gradually constructs a causal model of the task. Also Eyferth et al. (1982) hypothesized that participants gradually construct a system representation and connect it to existing schemata. This representation can take the form of structural knowledge about the system. Structural knowledge describes the functional or causal relationships between variables (Buchner, 1995). This knowledge is the result of hypothesis-formation and hypothesis-evaluation processes which progress through various phases. In the first stage knowledge is restricted to the pure identification of a relation between at least two variables (relational knowledge). During the second stage a more differentiated view allows statements about the direction (positive or negative) of the relation (sign knowledge) and in the third stage the precise amount of influence can be specified (numerical knowledge).

But in complex situations the problem solver does not always have access to all the relevant information and hence it is necessary to interact with the system to create the necessary mental models to deal with the task. According to Brehmer (1987), non-transparent microworlds place particularly high demands on participants' ability to generate and maintain mental models of the task for successful control. Using an ATC task Ackerman studied the process of skill acquisition through the three phases previously mentioned in section 2.2.3.1. It was found that, if the levels of transparency are high, participants arrive at the automatic stage, but low transparency hampered this transition. In another study (Putz-Osterloh & Lüer, 1981) it was found that when participants received a graphical representation of the relations among the system variables as an aid to increase transparency, the correlation between system performance and intelligence test measures was improved. In a non-transparent situation, where both the given and goal state and also the barriers to overcome it are non-transparent, it is difficult for a problem solver to evaluate his or her progress toward a problem solution, making it necessary for the problem solver to select and structure his or her interactions with the task in such a way that helpful information for evaluating progress can be extracted.

#### 2.3.2.3. Dynamics

The state of dynamic microworlds changes both autonomously and as a consequence of the decision maker's actions. This characteristic imposes a particular set of demands on the problem solver and shapes the problem solving situation, often making the opportunistic execution of an action in certain situations more important than the specific type of action (Omodei & Wearing, 1995). For example, in FireChief it is a good option to send a unit to drop water over a fire of low intensity that has just spread to a new cell, but if the decision is delayed the fire may increase in intensity, surpassing the capability of the unit to extinguish it. Research has found a negative relation between the level of dynamics and performance, for instance, an increase in dynamics yielded a decrease in the quality of system control in the SINUS microworld (Funke, 1993). An explanation is that the need to perceive and remember data while controlling the status of different units complicates the task and can hinder performance (Gonzalez et al., 2005). Decision making behaviour was considerably affected by dynamic aspects of the task environment in another set of tasks (Kerstholt, 1994) reinforcing this view. What is important is that the nature of the CPS situation changes as time elapses, becoming increasingly complex and by studying strategy use it is possible to determine whether a strategy is appropriate on a scenario-by-scenario basis (Schunn, McGregor & Saner, 2006). The study by Cañas et al. (2005), which is described in section 3.1, uses an experimental design that stresses the importance of studying how strategies change, or not, as environmental changes are introduced.

#### 2.3.2.4. Time pressure

A clock-driven simulation (in contrast with an event-driven one) does not wait for participants to make their inputs, thus increasing time pressure on the problem solver (Brehmer, 1995). Time pressure is an important factor relating to dynamic microworlds. In this kind of microworld, decisions have to be made in real time. As mentioned before, the morphing nature of the task makes it possible that a good decision at one moment may be obsolete in the next. From the perspective of the problem solver, there is evidence that the speed of information processing is increased as time pressure increases (Maule & Mackie, 1990; Payne et al., 1993). Research on military decision making (Thunholm, 2005) also stresses the importance of being fast and suggests that a cost-benefit perspective provides a good explanation of observed behaviour. Time pressure can yield to behaviours such as filtration (where less information is considered when making a decision) or acceleration of processing (Payne et. al, 1993). Several studies of strategy make use of combat scenarios (Schunn, McGregor & Saner, 2006) as this kind of task usually presents a high level of time pressure. Insights about strategy use under pressure can also be extracted from the Command and Control literature (Alberts & Hayes, 2002; Kuylenstierna et al., 2004; Thunholm, 2005).

#### 2.3.2.5. Multiple goals

The presence of multiple goals is another characteristic of microworlds. In a typical situation the problem solver is given an ambiguous goal definition, and s/he must derive a practical set of objectives. For example, in Moro, the problem solver receives instructions to "Improve the welfare of the Moro Tribe" and participants respond in many different ways: some of them focus on digging more wells, others on increasing the number of animals or improving health conditions, and so on. During interaction with Moro, players frequently face the situation of having conflicting goals. In a study about the relation of multiple goals and feedback delay using a fire-fighting microworld (Brehmer, 1995) the problem solver was given two goals: to protect the base and to put out the fire(s) as soon as possible. It was observed that participants learn to give a higher weight to the goal of protecting the base only when there are feedback delays. This result suggests that feedback delays make participants concentrate on the most important goal. This focus on protecting the base could be due to the fact that the feedback is less ambiguous with respect to the goal of preserving the base than with respect to that of extinguishing the fire as quickly as possible. Another explanation could be that participants underestimate the danger of the fire that is not threatening the base (Brehmer, 1995). In the ORTS microworld (Buro, 2000) the objective is to destroy all opponent buildings within 20 minutes. In this game several players fight over resources, which are scattered over a terrain, by first setting up economies, building armies, and ultimately trying to eliminate all enemy units and buildings (Buro, 2004). In this simulation participants can be forced to change their goal of harvesting resources when presence of enemy units is detected. In all these cases participants need to choose among different paths of behaviour and somehow evaluate the effectiveness of their selections.

### 2.3.3. Comparison of problem solving tasks

Figure 2.8 shows a three dimensional graph depicting the categorical location of the first five tasks described previously. Each dimension represents a chief characteristic described in section 2.3.1. Complexity is located on the x-axis, the y-axis represents dynamics and the z-axis represents non-transparency. As can be seen in the graph, the TOH (TH) and the F&T (FT) tasks have a single component (complexity), whilst the other three tasks present all three components. This distinction is important because, as mentioned previously, studies using the European approach to the study of CPS mainly use tasks that have the three components shown in figure 2.1 (section 2.2) whilst studies using the American approach do not. Coldstore, Moro and FireChief are considered microworlds. The addition of dynamics and non-transparency to the tasks poses a new set of demands on individuals in comparison to tasks that vary in complexity only. The graph also shows that Moro has the highest level of complexity while FireChief has the highest level of dynamics. Moro presents greater complexity as it involves more variables, more dependencies among them, and a wider range of actions. In FireChief the fire is developing continuously and in Moro variables are changing as simulation steps take place. FireChief therefore has the higher level of dynamics because it is clock driven rather than event driven. Güss et al (2004) found that participants consider a fire-fighting microworld similar to FireChief as more dynamic and complex than Coldstore.

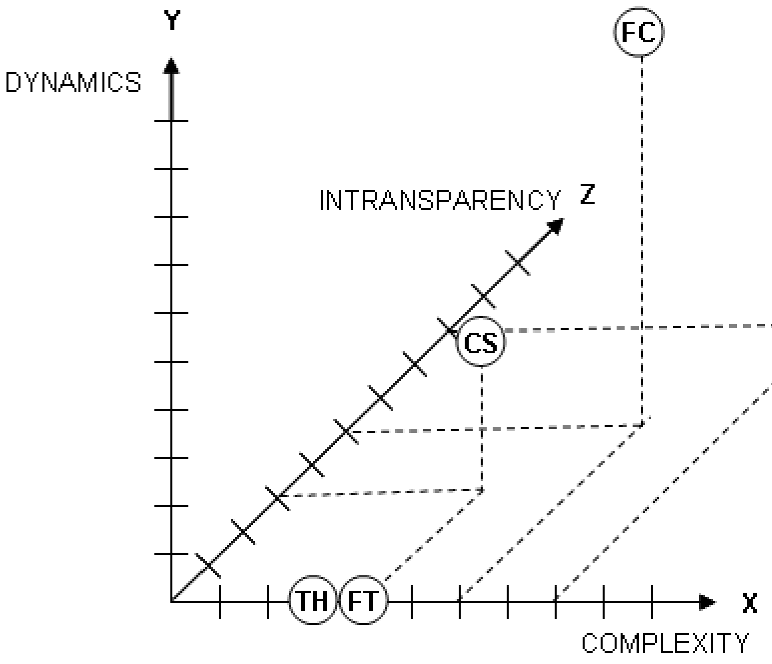



**Figure 2.8: Categorical distribution of different classes of microworld**

### 2.3.4. Methodological advantages of using microworlds

Microworlds allow for an economic and standardized presentation of complex scenarios that support research into problem solving behaviour. For example, FireChief is fully customizable and allows the definition of scripts for running of experiments (Omodei & Wearing, 1995). It is possible to simulate various years in the life of the Moro tribe in a single session with the results of decisions appearing immediately (Dörner, Stäudel & Strohschneider, 1986). Microworlds also provide an efficient means of data registration. The usual inputs from participants come from the keyboard and the mouse and can be logged along with application-related data. In the Kanfer-Ackerman Air Traffic Control task (described in the next chapter) sequences of strokes are stored including their duration (Ackerman, 1988). In addition, Brehmer & Dörner (1993) have observed that microworlds have acceptance from the participant's point of view, owing to their game-like nature. Simulations also provide the opportunity for practice and learning by doing (enabling the acquisition of declarative and procedural knowledge). They also provide different levels of feedback and their complexity can be adjusted at will (Funke, 1991, 2001). For this research the most interesting feature of microworlds is that their dynamic nature poses new requirements on participants in comparison to standard task applications, enabling the exploration of different cognitive abilities (Brehmer & Dörner, 1993; Gonzalez et al., 2004).

### 2.3.5. Methodological challenges using microworlds

There is a set of methodological challenges to be considered when microworlds are used for research. Computerized scenarios tend to produce a lot of behavioural data whose psychological interpretation may be unclear. Information generated by microworlds can be compressed into other measures and indices, for example, a matrix of transitions between actions can be obtained for a FireChief trial (see section 3.4.2). A common approach for analysing microworld data is to perform a task analysis to identify the required steps in the problem solving process (Ackerman, 1988; Taatgen, 2004; Dörner, 1996; Schunn & Reder, 2001; Rehling et al., 2004; Joslyn & Hunt, 1998, Kerstholt, 1996; Gray & Boehm-Davis, 2000). The researcher must look for useful indicators, those related to performance and timing being the most popular. An important piece of information used for interpreting results is the definition of the best solution for a problem. If the best solution is known, it is possible to determine the quality of a problem solving activity with more precision. In the TOH and F&T tasks it is possible to identify a precise sequence of actions to arrive to the solution in the least number of steps. In microworlds, the process of finding the best solution is harder due to two main reasons: the dynamic nature of microworlds and the ambiguity of goals. In FireChief two participants can issue the same actions for an identical trial and yet obtain different results. The reason is that the temporal dimension is quite important in this microworld: the fire is destroying the landscape at all times. Two participants will only produce the same results if the sequence of actions are executed in the same spatial and temporal sequence and, considering the variability inherent to the execution of cognitive, perceptual and motor actions (a subject discussed in the cognitive architectures sections) it is unlikely that two participants will present the exact same pattern of actions. The ambiguity of goals also makes harder the definition of an optimal solution. In Moro the goal is to promote the welfare of the tribe; in this case the definition of welfare is ill-defined. Some participants may concentrate on providing medical attention to the population while others on increasing the amount of fertile land. In both cases participants are contributing to the welfare of the tribe but, without a clear definition of an optimal solution, participants may concentrate their efforts in following a path that is not considered optimal by the researcher’s definition of the optimal solution.

There is a dilemma concerning experimental control related with the use of microworlds. In microworlds the stimulus is no longer under the full control of the experimenter but depends upon the initial configuration of the simulation (defined by the experimenter) and the sequence of interactions executed by the participant (Gonzalez et al., 2004). In other words, participants in the same control group do not receive the exact same treatment, and this is a violation of the experimental control assumption. In this scenario it is hard to compare participants' behaviour. This problem can be tackled from two perspectives: increase experimental control or modify the analytic approach. Funke (1998) proposed the reduction of stimuli variance whilst taking into consideration that strict control of the simulation is not compatible with the use of microworlds. Streufert et al. (1988) applied this approach in a study about the relation between managerial performance and training. In this work Streufert et al. make a distinction between free simulations, which are extremely dependent on participants’ actions, and quasi-experimental simulations in which the experimenter has control over the presentation of information. In a quasi-experimental simulation “each participant receives the same quantity of information at fixed points in simulated time” (p. 5). Dörner proposed an alternative solution that respects the simulation principle: focus the analysis on patterns and strategies. According to Schunn, McGregor & Saner (2006) a strategy is a coherent set of steps for solving a problem. Brehmer & Dörner (1993) say that in fire-fighting experiments stable results should be obtained at the levels of goal attainment and strategy rather that at the level of individual decisions; this approach is followed by various researchers (Cañas et al., 2005; Taatgen, 2005; Schunn & Reder, 2001).

### 2.3.6. Dealing with microworlds

In order to successfully deal with microworlds a problem solver can make use of a set of mechanisms to maximize his/her cognitive resources utilization. These mechanisms are explored next as they will drive the construction of the cognitive model.

#### 2.3.6.1. Choose an appropriate representation of the problem

A task representation includes a subset of all possible task features (Schunn & Reder, 1998). The ultimate goal of this representation is to enable to problem solver to gain control over the simulation. Niessen et al. (1999) stress the importance of coordinating activities in a dynamic environment by means of a mental representation. They identify three processing requirements for enabling situation awareness: the update of varying relations, the prediction of future states, and the determination of the temporal sequence of cognitive activities. They developed a model following these principles for an ATC task called EnCoRe that simulates traffic scenarios providing radar screen runs, electronic flight-strips, and headphone communication with a ghost-pilot. One of their studies investigated the selection of information and involved masking sections of the radar and the flight-strip with grey bars. It was found that "…the representation of the traffic situation is built up by means of a considerably reduced amount of information" (p. 9), that is, participants were able to build a sufficient representation of the situation by focusing on just a few aspects of the task. From the 6 available sources of information 3 were preferred over the rest: *callsign* (to identify the aircraft), *Flightlevel* (vertical position of aircraft in space) and *predictor* (flight direction). Other sources of information such as *speed, cleared flightlevel and c/d indicator* were not preferred by problem solvers. The authors argue that the callsign information is used to identify the aircraft and together with the Flightlevel and Predictor indicators a spatial representation is constructed.

For some problems the difficulty resides in finding the right representation, because some representations might involve much simpler processing operations than others. Much research on expert-novice differences highlights experts' great advantage in properly representing and categorizing problems in terms of deep features (Demetriou et al., 2002). A difference between novices and experts found by Dörner (1996) with the microworld Moro is that the former tend to ignore side effects during interaction. Another difference is that novices focus on surface features while experts focus on deep features. In this respect Sloutsky & Yarlas (2000) studied novice-expert differences in the construction of arithmetic equation representations. Their experimental design consisted of a learning phase where items which present both superficial and deep features were presented. Deep features include deep relational properties such as principles of commutativity and associativity while surface elements include commonality of numbers or the same number of constituent addends in the equation. In a second phase, participants tried to recognize the items. After considering different processing models, they found that novices and experts represent both types of features but that there is an attentional competition between surface and principled features for novices. Anderson & Reder (1998) also observed that novices tend to lack the relevant knowledge structures to classify problems as well the experts, and that experts represent the current knowledge state with fewer 'chunks' in comparison with novices. In this sense, experts and novices have different ways of selecting strategies and adapting to the environment. The process of structural knowledge acquisition can also be influenced by the semantic context of the task. For instance if participants are given instructions that frame a task with a familiar context they tend to show a confidence level that prevents them from properly exploring the system and hence obtaining useful structural knowledge (Beckmann & Guthke, 1995). Because the system variable most strongly related to the creation of a system representation is the level of transparency (Hörmann & Thomas, 1989); the less transparent the system (such as in the case of microworlds) the harder it is to develop a representation of the system.

### 2.3.6.2. Implement strategies

Regardless of the high complexity of microworlds, researchers have been able to successfully identify and describe strategies when conducting experiments. Lee, Anderson & Matessa (1995) studied different strategies using the KA-ATC task (see section 2.3.1.7). They observed an improvement in performance over trials and tried to explain it based on strategy change. Landing an aircraft typically involves progressing from the landing queue through 3 hold levels where level 1 immediately precedes landing. They identified a common strategy among participants which they called 'hold 1'. This strategy involves bringing aircraft from the queue directly into priority hold level 1 (skipping two hold levels). The hold 1 strategy reduces the number of keystrokes needed to land an aircraft. This measure is defined as the percentage of aircraft brought directly from the queue into hold level 1. It was observed that this strategy was used increasingly until half way through the experiment and after that it's used all the time. Another strategy was to maximize the number of aircraft landing simultaneously. When the wind changes direction, there is an opportunity to use the runways in the new direction while other aircraft are taxiing. They called this second strategy 'runway efficiency' and observed that the frequency of this strategy increases throughout the experiment. They also added two psychomotor variables: the number of keystrokes and the mean reaction time to wind direction change; together with the two measures of strategy and the score this summed up to give a total of five measures. From these measures, runway efficiency correlated most strongly with the overall score, from which the authors conclude that scores increase with the adoption of either the *hold 1* or the *runway efficiency* strategies, and that overall speed also contributes to the score. The authors conclude that the strategy use of participants contributes significantly to performance and suggest that any model of this kind of task should be able to learn over trials which strategies are better and how to execute them more efficiently.

Charman and Howes (2002) were interested in the effect of practice on strategy generation. Their study was focused on the tendencies of people when selecting between known strategies and the cognitive cost of learning new strategies. The task used was a *drawing task* where participants needed to either design the layout of a computer classroom and a study area (the higher-goal condition) or to copy series of elements into a blank area (the no-higher-goal condition). In both cases, the items to be arranged are 148 computers and 54 desks. Three main strategies were identified: work with individual shapes, work with more than one item at a time by making multiple selections, and use the exponential copying strategy where an initial element is copied, then the resulting two elements are selected and copied producing four elements, and then these four are selected and copied an so forth. The results show that participants with higher constraints are able to generate the most elaborated strategies observed in the no-higher-goal condition group, but generate them later in time due to the reduced opportunity to practice more efficient strategies (such as exponential copying) due to the existence of a higher goal. The conclusion of the experiment was that, as users become more familiar with the experimental environment, better use of its operators is enabled (and a shift in strategies takes place).

Gray et al. (2005) studied strategic selection using the Blocks World Task, where the objective is to copy a pattern of blocks. They identified a trade-off between the accuracy of a strategy and the effort required to execute it. They divided the task into elementary information processes (fundamental cognitive operations). The experimental manipulation was a variation in the cost of accessing information in one experimental condition including hiding certain information and using delays. They found that participants are responsive to these manipulations and adapt strategy use to them: members of the group where there was a delay uncovered the information fewer times. Individual differences in strategy choice were also studied by Ball et al. (2003) using an ATC where participants fly seven distinct manoeuvres and their performance is obtained by calculating the RMSD of each trial with the ideal performance. The authors were interested in studying differences in knowledge and strategies. They used a cognitive modelling approach in which task different variants of the model are tested. Each new variant of the model incorporates more features than the previous one. In this task there are two types of indicator. Performance indicators reflect the behaviour of the aircraft (e.g. airspeed and altitude) and control indicators reflect the settings of the controls which affect the behaviour of the aircraft, such as engine speed. Particularly they compared two strategies: attending to control and performance indicators vs. focusing on control indicators only. The first strategy implies a crosscheck between performance and control indicators while the second allows the correct setting for control instruments. They found that the second strategy was more successful in five out of seven trials mainly because this second strategy allows a higher number of quick adjustments to airspeed, altitude and/or heading allowing better performance. Differences in strategy were also detected in the Emergency Dispatch task (Joslyn & Hunt, 1998). Participants who gave preference to higher priority incidents lost points mainly on low priority incidents.

Implementing a strategy in a complex dynamic task also places significant psychomotor demands. The problem solver must issue a considerable amount of commands to the various objects in the simulation for achieving his or her goals, and doing so under time constraints. In a study using a different ATC task called AMBR, Rehling et al. (2004) studied individual differences. The task is similar to the KA-ATC task but uses radar to display aircraft locations. In this experimental design, psychometric tests were applied for assessing two cognitive abilities, working memory and psychomotor capacity, and an ACT-R cognitive model of the task was parameterised based on these same parameters. The authors report that, although measures of both cognitive abilities are able to predict performance on the AMBR task, psychomotor ability is a better predictor. As the ATC model developed by Anderson et al. (2004) gains experience motor ability becomes more relevant. This result provides more evidence of the importance of psychomotor ability in dynamic tasks.

All the studies discussed in this section have one thing in common: their focus on the strategies participants use to deal with the task at hand. Different aspects of strategy use were considered in this section: strategy identification, strategy choice, and the effect of practice on strategy choice. All these elements are further explored in the following chapters. Also different ways of analysing strategies (for instance, by decomposing the task) and measuring their impact on performance or other relevant measures have been presented here: the following chapter describes how a brand new set of strategies were found for a dataset obtained using the FireChief microworld.

#### 2.3.6.3. Use WM efficiently

Studies using microworlds have found that performance is related to measures of cognitive ability (Brehmer & Dörner, 1993; Taatgen, 2001, Rehling et al., 2004). For example Rehling et al. (2004) found a positive correlation of working memory (WM) and psychomotor ability with performance in an Air Traffic Control (ATC) task. SüB et al. (2002) identified various functions of WM such as its influence over individual ability to simultaneously store and process information, required for the performance of supervisory functions (such as monitoring mental operations and controlling their efficiency). Schunn & Reder (2001) argue that, while performing the KA-ATC task, WM is necessary for keeping information in mind (the rules for landing aircraft) to allow the appropriate selection of a runway (long or short) for landing the aircraft. In general terms a successful interaction with complex, dynamic tasks requires tracking the development of environmental variables such as the aircraft that have been classified (Ackerman, 1998), the current emergency report that is being attended (Joslyn & Hunt, 1998), or the strength of the wind (Schunn & Reder, 2001), and storing this information places demands on WM.

Brumback et al. (2005) stress the importance of working memory for controlling the allocation of attention. They grouped participants according to their performance in the Ospan task (LaPointe & Engle, 1990) and compared their performance in a couple of auditory tasks involving mapping tones and specific buttons. Results show that participants with low-Ospan (4-9) scores are more distracted by changes in the stimulus than participants with high scores (19-41). Brumback et al. conclude that the capacity to control attention is important in tasks that require complex cognitive processing, such as the FireChief microworld. Gonzalez et al. (2004) concluded that FireChief is characterized by its dynamic nature and that participants show more dependence on WM because there is a lot of information and events to look at and therefore performance is associated primarily with the ability to store and process visual or spatial information.

#### 2.3.6.4. Focus on controlling the situation

Instructions given to participants interacting with microworlds are usually diverse and vague. Nevertheless, research has found that there is a common goal people have when interacting with a simulation: to gain control over the situation (Brehmer, 1992, Brehmer & Dörner 1993). Brehmer and Allard (1991) give an example of how one process can control another, this analysis can be applied to the FireChief task, where on one hand there is a fire fighting process which uses appliances for fighting the fire and on the other hand there is a process that represents a developing fire. In the case of a fire burning in a uniform environment (e.g. a forest), the size of the burning area increases linearly with time and the rate at which it spreads depends on the nature of the forest and the strength of the wind. The effect of the fire-fighting process also increases linearly with time; each fire-fighting appliance will be able to cover a constant forest area with water per unit time.

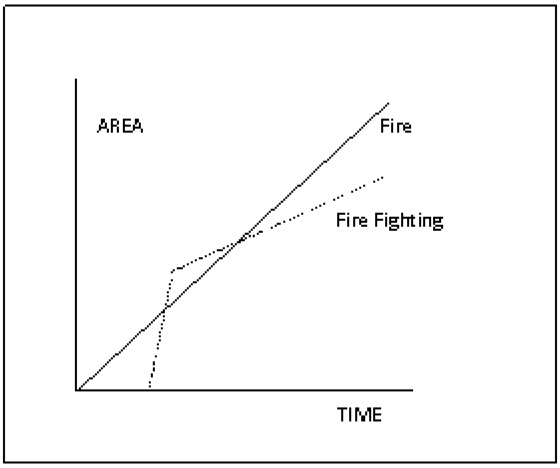



**Figure 2.9: Diagram of the Fire and the Fire-fighting processes taken from Brehmer and Allard (1991)**

Figure 2.9 shows the development of two processes involved in a CPS situation. The fire process is active from the beginning of the simulation (fire is burning and destroying cells when the trial starts) but the fire-fighting process is not: a period of time is necessary before starting to fight the fire (the situation should be evaluated, a course of action selected after which the units can move and start fighting the fire); this period of time is known as 'dead time'. The critical point is where the two processes intercept. According to Brehmer and Allard (1991), before the first interception, the fire can be extinguished by direct attack, as the fire-fighting units are able to cover at least the same area that the fire is covering, but after the second interception, the best course of action is to stop the fire from spreading using fire-control. Spatio-temporal diagrams such as that in figure 3.1 can show the strategic options FireChief participants have (Brehmer, 1995). The assumption that participants will attempt to control the situation at hand (1) limits the set of possible actions/intentions and (2) favours those actions/intentions with higher possibility of achieving this control; these features are useful when the cognitive modelling approach is followed (described in chapter 5).

### 2.3.6.5. Adapt to changing environment

When participants interact with a complex, dynamic and non-transparent microworld two kinds of control can be exerted: top-down (by following a plan or a strategy) and bottom-up (by reacting to task events). In the words of Payne et al. (1988) "… a decision maker possesses a repertoire of well-defined strategies and selects among them when faced with a decision by considering the expected costs and expected benefits of each strategy" (p. 18). A strategy considered as a form of top-down control implies that the current representation of the task has an influence on the actions that are executed when interacting with the microworld. Section 2.3.6.2 showed the importance of implementing the right strategy in order to obtain good performance in a variety of tasks. Nevertheless bottom-up control is particularly relevant when considering dynamic tasks. For Bettman (1979) strategies develop in an ad hoc fashion during the course of the problem solving process, and these strategies are generated by being receptive to the environment. Bottom-up control is driven by feedback obtained from the environment and is used for guiding the selection of actions; the outcome of this process is adaptation.

Adaptation is also related to the ability to monitor your own strategies. The term metacognition is often used for this kind of cognitive activity (Metcalfe & Greene, 2007). Returning to the topic of the importance of gaining control, stressed by Brehmer (1992) in section 2.3.6.4, FireChief participants should continually ask themselves if they have the fire under control, that is, if their strategy is able to control fire development. In order to be adaptive, cognitive resources must be used for this end and, as Brehmer (1992) pointed out, there is a compromise between a strategy for controlling the task and a strategy that gives the problem solver control over the rate at which he has to make decisions. One of these decisions is whether a particular strategy should be changed. This topic is analysed further in section 5.4.4 (sub-section 15).

One of the research questions of the present work is how do choices arise in complex and dynamic situations. The answer to this question seems to be related to a combination of using strategies and being receptive to environmental cues. Payne et al. (1988) argue that a major problem is to understand and predict when a particular strategy will be used. If we focus our attention on the concepts of effort and accuracy and map them to ACT-R concepts we find first that in ACT-R effort is measured as the time associated with the firing of the productions related to a specific strategy. The time required for each production is a function of the actions executed by it: cognitive, perceptual or motor. And second, that accuracy is a function of the feedback received from the environment in the form of rewards. Chapter 5 describes these concepts in depth as they are critical to understanding behaviour during the running of the model. In FireChief, for example, accessing information requires a combination of cognitive and perceptual actions that are executed by the problem solver and have a cost that can be measured in time. In making a decision about how to use a fire-fighting unit (the most important resource) a participant can either follow a plan, and use a less perception-intensive approach, or base his or her choice on what is perceived from the environment. In the second case the problem solver needs to determine how much information is going to be gathered before making a decision. As the problem solver interacts with the FireChief task he/she learns to make this kind of decision based on their performance.

For Lovett & Schunn (1999), as experience in the task increases, an individual will move towards a strategy representation that includes performance data and will use the strategy that produces the highest success rate, so a strategy will be used more often if it is successful. For Lovett & Schunn (1999) the problem solver's task representation (a set of stimulus features) modulates his or her ability to learn base-rates when these base-rates are learned by experience. In this sense, a representation acts as a filter that determines what can be learned from interactions with the task (Lovett & Schunn, 1999).

## 2.4. Cognitive Architectures

Cognitive architectures have the goal of providing a framework capable of enabling the versatility observed in human behaviour. A cognitive architecture embodies structures and mechanisms in the form of a general theory of how the mind works (Newell, 1990) and is used for creating simulation models of human cognition (Taatgen & Anderson, 2008). Cognitive architectures are implementations of what Newell called *Unified Theories of Cognition* (UTC). An UTC strives to cover a large number of domains (a single architecture should be used to account for as much regularity in empirical data as possible) and to consider multiple constraints in order to reduce the number of degrees of freedom in the theory that is being studied (Newell, 1990). The emergence of cognitive architectures was possible mainly due to two factors (Newell, 1987). On one hand psychologists shaped various theories about aspects of cognition such as memory, problem solving, perception and psychomotor actions (among many others). And on the other hand developments in computer-related technologies, such as more robust and faster hardware and better algorithms, allow the execution of increasingly complex computations. These improvements offer the possibility of implementing computationally demanding psychological theories in increasingly shorter times. The advantages of modelling psychological processes in computers are similar to the ones described in the microworlds section, namely, the economic and standardized generation of data (the model is able to interact with computer tasks in a similar fashion as human participants) and, mainly, the fine-grained level of detail in which the phenomenon of interest is captured.

The rich and variable behaviour of humans can be divided into different bands based on time scales (Newell, 1990). Each band has its own set of operations that interact with each other to produce behaviour. The bottom level corresponds to the biological band; it represents the neural level and its operations are of the order of 1 ms. The biological band also includes the neural circuit level that uses activation levels of the order of 10 ms. At this level symbols correspond to patterns of neural signals. The existence of a biological band imposes a real time constraint on the cognitive band (the second layer) of the order of 100 ms. A sub-level within the cognitive band is called the deliberation level where an agent deliberates about when to use knowledge to choose among options. Deliberation requires three serial steps: knowledge retrieval, decision and execution, each of them of the order of 10 ms. This kind of deliberation is simple and is considered to be automatic. Most theories consider that cognition operates on a time scale of 10 ms. to 10 seconds. Above 10 ms. the cognitive band is capable of deliberate acts and operations (Anderson, 2002) and hence differences in behaviour need an explanation based on the task or particular architectural or epistemological features of the problem solver (or a combination of them). When operations last more than 10 seconds the system is operating in the rational band. In this band the agent applies all his available knowledge to obtain his or her goals. Regarding the time scale, the rational band encompasses tasks of the order of minutes to hours. There are other bands at a higher level that are not discussed here. In this respect Anderson (2002) stresses the importance of spanning the order of magnitude of cognitive models, that is, to create models of cognition that offer explanations of phenomena beyond the cognitive band. Returning to the definition of cognitive architectures and linking this to time scales, architectures are fixed definitions of the layer below the symbol level (Anderson, 2002) where more complex operations depend upon the latencies of the lower layers, and these latencies can be changed by accumulating experience. Cognitive architectures constrain modelling and should only allow cognitively plausible models (Taatgen et al., 2006).

### 2.4.1. Overview of the ACT-R cognitive architecture

This section focuses on describing the cognitive architecture used in this research: ACT-R (Anderson et al., 2004). This discussion gives particular attention to how actions are selected, the focal point of this research. There are other cognitive architectures such as Soar (Laird, Rosenbloom & Newell, 1987), EPIC (Executive Process-Interactive Control, Kieras & Meyer, 1997) which couples perceptual-motor mechanisms with a production-system and Cogent (Cooper & Fox, 1998). All of these architectures are capable of realizing intelligent behaviour. This discussion is enriched with comments of how cognition is realized in Soar.

A cognitive model's performance is intrinsically related to the cognitive architecture in which it is implemented and executed. The decision of selecting ACT-R is constrained by the CPS paradigm depicted in figure 2.1 in which there are two main elements: the problem solver and the task. In the case of the problem solver it is important to have a way of modelling different aspects related to strategy use; whereas in relation to the task the model must be able to deal with a complex, non-transparent and dynamic environment. In this research it is hypothesized that learning generated by feedback from the environment is particularly relevant to achieving good performance in a dynamic microworld such as FireChief. For this reason, the most important criterion for selecting a cognitive architecture was its ability to process feedback from the environment. The reinforcement learning mechanism in Soar creates rules and it adjusts the values of preferences for operators. On the other hand in ACT-R the sub-symbolic calculations modify the utility value of every rule. That is, ACT-R tunes the utility value of every single production according to environmental feedback. This learning mechanism allows an ACT-R model to adapt its selection of rules at a fine-grained level. This is particularly relevant when dealing with a highly dynamic task such as FireChief.

An important difference between these cognitive architectures is in how performance limitations are explained. In ACT-R there is a central-processing bottleneck: only one rule can be fired at a time. The serial bottleneck in ACT-R limits the amount of parallelism. Soar has a more arbitrary way of matching patterns and performance limitations are explained more by lack of knowledge. Instead of making several elaborations (in parallel) before selecting an operator (for example, to execute a particular FireChief command) or switching between problem spaces, an ACT-R model fires a single rule at the end of each cycle. Considering the complexity and dynamic characteristics of FireChief, the number of elaborations made by Soar in parallel would escalate to such an extent that the analysis (and comprehension) of the model's behaviour would become highly difficult. The parsimonious approach followed by ACT-R allows the exploitation of the utility learning mechanisms for analytical purposes (see sections 5.2.3.2 and 6.1.2.3). As pointed out by Taatgen & Anderson (2008) for Soar rationality refers to using available knowledge for dealing with the task whilst in ACT-R rationality is to optimally adapt to the environment. In Soar it is rational to use all the available knowledge. In the context of FireChief this knowledge includes, among other things, the position of the different fires, the location and status of the fire-fighting units, and the current strategy. Soar would use all this knowledge for making this decision. On the other hand, an ACT-R model may choose to use less knowledge for making this decision if the cost of acquiring more knowledge is high.

The characteristics of the ACT-R visual module are another reason for choosing this cognitive architecture. The ACT-R “visual interface” is designed to enable the modelling of dynamic, high-performance perceptual-motor tasks. ACT-R can also issue keystrokes and execute mouse commands and is able to handle movement and change, can be instructed to track moving objects, can adjust to very small movements, and can correctly identify objects in changed displays, just the kind of abilities required for interacting with dynamic tasks. In FireChief visual elements change frequently. Visual elements can be perceived by the model and this process creates facts in the memory of the model which are used to make a variety of decisions. ACT-R is also capable of hearing simple stimuli which is necessary for interacting with complex task domains such as FireChief. ACT-R can model individual differences in cognitive ability by manipulating architectural parameters. The ACT-R architecture contains a number of parameters that can be used to fix levels of performance (Rehling et al., 2004). This approach to modelling individual differences has been used in many ACT-R models. For example, differences in psychomotor speed have been modelled by varying the time needed for a key-press (Taatgen, 2001), or the speed of proceduralization by controlling the learning rate (Rehling et al., 2004). The basic principle of ACT-R is that an agent executes actions according to *rational analysis*: it selects actions trying to achieve its goals. In this way the ACT-R cognitive architecture is constrained to follow a goal and to prefer those actions that are more likely to obtain positive feedback from the environment. Similarly SOAR is guided by *the principle of rationality*: if an agent has knowledge that an operator application will lead to one of its goals then the agent will select that operator (Newell, 1982).

#### 2.4.1.1. Architecture

The ACT-R (Anderson et al., 2004) architecture is composed of several specialized modules interconnected by channels called buffers. The central block in figure 2.10 shows the procedural module where procedural knowledge is stored in the form of production rules. This module implements the procedural memory and connects the rest of the modules (Taatgen et al., 2006). The visual module searches and identifies objects in the visual field, the manual module controls the hands, the declarative module retrieves information from memory, and the goal and imaginary modules keep track of the model’s current intention and the context of the problem solving situation respectively. Communication between modules is achieved through buffers; the content of any buffer is limited to a single declarative unit of knowledge called a chunk (section 2.4.1.2). For this reason the system can only be aware of a limited amount of information at a single moment in time. This last constraint is inspired, for instance, by the fact that people are not aware of all the information in their visual field but only the object they are currently attending to, and people are not aware of all the information in long-term memory but only the fact currently retrieved (Anderson et al., 2004). The central production system can only execute one rule at a time. ACT-R processes requests for a resource located in any of its modules on a first-come, first-served basis (Taatgen, 2005). Each module is mapped to the actual human cerebrum. For example, the goal buffer is associated with the dorso-lateral prefrontal cortex (Fincham et al., 2002) but also with other regions.

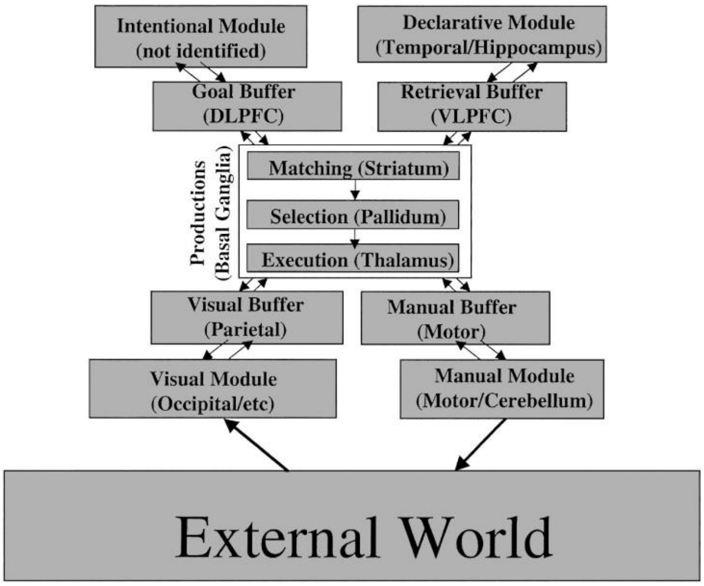



**Figure 2.10: ACT-R architecture, taken from Anderson et al. (2004)**

### 2.4.1.2. Types of knowledge

ACT-R makes a distinction between declarative and procedural knowledge. Anderson & Lebiere (1998) propose that chunks and productions are the atomic components of thought because they represent the lowest unit in the symbolic decomposition, that is, below chunks you only have unrelated combinations of values and below rules you cannot define steps of cognition. Declarative knowledge is represented by chunks, which are elements that encode things we know such as: "the wind is blowing to the right". Procedural knowledge is represented by rules, and can be used to produce task-specific behaviour. Procedural knowledge describes how to do things, for instance it may specify: "to find a fire in the landscape look for a cell that is red". There is evidence of this distinction in the brain: the basal ganglia are related to the processing of procedural knowledge and the area of the hippocampus to new declarative knowledge (Anderson et al., 2004).

In the ACT-R theory, the goal of human memory is to retrieve appropriate knowledge to solve a current task. Requests to the declarative memory module retrieve a chunk from the declarative memory that matches certain criteria and place that chunk into the retrieval buffer (if found). Each declarative piece of knowledge has an activation level which determines its probability of being processed by procedural knowledge. As a result, activation ranks knowledge in order of potential relevance (Taatgen & Anderson, 2008). The activation formula has two main components, the base-level activation (general usefulness in the past) and associative activation (relevance to the current context). The base level component of activation fluctuates and decays with time allowing ACT-R to adapt to changing needs over time. Activation also has a random component which allows ACT-R to explore the usefulness of different facts and a partial-matching component that allows rules to fire even though not all conditions are matched. Bothell (2007) describes all these elements in detail.

Each rule, the unit of procedural knowledge, corresponds to a step in cognition and is comprised by two elements: a condition and an action. The condition of a production rule (the *if* part) is a collection of patterns to be matched (for instance, this pattern can correspond to the content of buffers in ACT-R) and the action of a production rule (the *then* part) consists of the execution of one or more transformations. ACT-R rules are in charge of querying and transforming the content of buffers and these transformations represent cognitive steps. There are specific constraints such as the use of one memory retrieval per production and the control of discrete behaviours (such as key press, eye movement, and generation of an utterance) by distinct productions.

### 2.4.1.3. Basic Cycle

During the basic rule-firing cycle in ACT-R the various buffers hold representations determined by the current task and internal events. Production rules recognize this set of representations or pattern and a single production fires, the firing of this production may update the content of one of or more buffers therefore activating other rules (50 ms. represents the minimum cycle time for processing). ACT-R uses the Rete algorithm (Forgy, 1982) for the pattern matching process. There is a latency associated with the firing of every production which is calculated by accumulating the time required for performing all the actions specified in the rule, such as retrieving chunks, moving the pointer of the mouse, hearing a sound, etc. Parallel processing exists if actions are executed by different ACT-R modules, such as moving attention and pressing a key in the keyboard or moving the eyes and the mouse at the same time.

### 2.4.1.4. Action selection

In ACT-R the next action is determined by the rule that is fired, nevertheless there may be a conflict resolution process when more than one production can be fired. Von Neumann & Morgenstern (1947) proposed that, in decision making under risk, a decision-maker should choose the course of action that maximizes the expected utility of the outcome. This principle is implemented by the ACT-R conflict resolution process, which is responsible for selecting the next rule to be fired from the set of rules whose preconditions comply with the current content of the different buffers (the conflict set). The productions in the conflict set are ordered in terms of their expected gain (i.e. utility) and ACT-R considers them according to that ordering (soft-max function) and a random component (noise). The formula used to obtain the utility of a production is shown below. When a reward is triggered it changes the utility of the most recent productions. The new utility of a production ($U_i(n)$) is a function of its current utility ($U_i(n-1)$) and the value of the effective reward ($R_i(n)$). The value of the effective reward decreases as the difference it time between the firing of the rule and the giving of the reward increases. It is important to note that there is a stochastic component during the process of conflict resolution. In ACT-R conflict resolution is adaptive and satisficing (it uses a threshold, and when the expected gain of a production exceeds it, it fires). It has been observed that satisficing rational agents do not select the most effective strategy for solving a task but rather set a specific aspiration level and select a strategy that exceeds it (Johnson, 1997). After a production fires, the pattern matching and conflict resolution processes will be performed again and this loop will continue until the model has finished (i.e. no production can be fired).

The following formula controls the learning of utilities:

$$U_i(n) = U_i(n-1) + \alpha[R_i(n) - U_i(n-1)]$$ **Difference Learning Equation**

$\alpha$ is the learning rate.
Ri(n) is the effective reward value given to production i at time n.
Ui(0) is set by a parameter.

It is worth mentioning that for Johnson (1997) the different strategies for conflict resolution are the major difference in control between ACT-R and Soar. ACT-R uses an automatic conflict resolution method described above whilst Soar uses a knowledge-based conflict resolution method based on all available knowledge. Making good decisions is a critical aspect of performance when dealing with simulations such as microworlds. Being one focus of this research how actions are selected, and how this selection is influenced by task characteristics and the feedback participants receive from the environment, the conflict resolution process and its most important determinant, the utility of productions, are of the utmost importance.

#### 2.4.1.5. Learning

In ACT-R there is learning at two levels: symbolic and sub-symbolic. ACT-R can learn new productions through the production compilation mechanism. In ACT-R there are special structures that must be explicitly created and tuned to accomplish the compilation of rules; these structures specify how rules can be compiled together to improve speed. Performance can be sped-up by reducing the amount of knowledge that needs to be retrieved. At the sub-symbolic level ACT-R can modify the utility of production rules in response to environmental feedback (reinforcement learning).

#### 2.4.1.6. Goal representation

In ACT-R the function of the goal module is to enable the model to respond differently to stimuli accordingly to a special structure called the goal. The goal determines the manipulations that the system applies to the available elements of the environment. In this way the firing of productions is organized into coherent behaviour (Anderson et al., 2004). The goal and imaginal buffers are considered as the working memory of participants (Veksler, Gray & Schoelles, 2007). A production must explicitly set a goal by placing a chunk in the goal buffer.

## 2.5. Cognitive modelling of complex behaviour

The most compelling reason behind the selection of FireChief was its salient dynamic nature. Anderson et al. (2004) propose the idea that dynamic tasks are ideal domains for testing the integration of ACT-R modules given that in these tasks is necessary to execute perceptual and motor actions whilst retrieving information from memory and all these processes are controlled by a central production system. This section starts by describing the technique used for developing this study: cognitive modelling.

### 2.5.1. Cognitive Modelling

As Newell (1990) points out, human activity implies an enormous variety of activities. Much of this behaviour is a consequence of cognitive processes which are not open to conscious experience making it necessary to develop methods to study it. One of these methods is cognitive modelling. Cognitive modelling refers to the modelling of higher-level cognition by means of computational technologies such as production systems (Anderson et al., 1995). Cognitive modelling does not share the AI goal of creating intelligent programs (Taatgen & Anderson, 2008). For this reason it is not enough to solve a problem: the solution must be cognitively and neurologically plausible. Because a computational approach (with explicit data structures and mechanisms) is used in constructing cognitive models, a detailed specification of the theoretical elements is mandatory (Cooper et al., 1996). For Jones, Ritter & Wood (2000) the development of models forces the definition of ill-defined concepts, such as cognitive inflexibility (described in the following chapter). Cognitive modelling can be used to obtain predictions based on theories by means of simulation (Cooper et al., 1996).

Various approaches have been used for developing cognitive models for tasks such as microworlds. For instance, for the Instance-Based Learning Theory *(*Gonzalez et al., 2003) the basic idea is that individuals acquire and store instances (an instance contains information about the situation, the action and the result of a decision) and then uses these instances to make decisions during subsequent interaction with a dynamic environment. This theory has an implementation in ACT-R called CogIBLT. For testing this model the Water Purification Plant task (a scheduling task where water is processed and performance is measured by the number of gallons not successfully purified) was used. It was observed that performance improves as the situation-decision-utility triplets are refined due to the environmental feedback. There are also some guidelines that can be use when developing models; for instance, when designing a cognitive model, Taatgen (2005) proposes the use of the *minimal control principle*: to use the least possible number of goal states to reduce/minimize brittleness. Avoiding brittle behaviour was a continuous concern during the development of the cognitive model for FireChief (see section 4.2.5).

### 2.5.2. Strategy use and cognitive modelling

As stated in Frensch and Funke (1995), a problem such as FireChief cannot be solved by "utilizing a small number of domain-general heuristics, but rather can only be solved by using domain-specific knowledge and procedures" (p. 12); these procedures come in the form of strategies. Because this research has the goal of increasing our understanding about the cognitive processes underlying solution strategies in complex dynamic tasks by following a cognitive modelling path, research related to this topic is presented here. Lee & Johnson-Laird (2004) suggested that research into strategy use should devise experiments able to reveal the solution strategy at detailed levels, including differences among participants and type of stimuli used in the experiments. They also suggested that, in order to investigate the development of strategies for solving problems, it is necessary to use problems that can be presented in a series that call for distinct solutions. The following chapter demonstrates that the selected dataset has those characteristics and that microworlds allow the intended experimental control; on top of this, cognitive modelling can provide a fine-grained level of description. The following examples show that considerable effort has been devoted to modelling strategy use in complex tasks.

Peebles & Bothell (2004) created an ACT-R model for the Sustained Attention to Response Task (SART). In this task participants are presented with the digits 1 to 9 in random order (one every 1.15 seconds). If participants see the number 3 they must withhold the response. The authors tried to identify the performance determinant in this task: either the ability to sustain attention or speed-accuracy trade-off. The authors also explored another possibility: strategic factors, that is, the ability to find an appropriate strategy for minimising reaction time and error. In their ACT-R model they allowed two strategies to compete. One strategy is faster and less accurate: the model clicks the mouse after detecting the stimulus; the other is more accurate but slower: the model checks the stimulus to determine what to do. The model consists of 11 productions and strategy selection in the SART task is controlled by utility. The SART model also has an explicit way of modifying its strategy. If the model detects that it responded correctly it sticks with the current strategy for the next trial, but if it found that an error was committed then it can either stick with the current strategy for the next trial or use a more accurate strategy. To fit the model Peebles and Bothell adjusted the *s* parameter (this parameter controls the amount of variance in the noise added to utility calculation) to .01 (a small value compared to the standard value of 3). The authors argue that a low value of *s* allows the model to be sensitive to changes in the utility values of each production and to show some variability. The SART study found evidence of an interaction between strategy use and performance. The authors consider that the ACT-R utility learning mechanism has a central role in the appropriate selection of strategies. Nevertheless, in the SART model the selection of strategies is also influenced by explicit symbolic processes: after detecting an error the model changes strategy.

Argus Prime (Schoelles & Gray, 2000) is a task that involves radar operation where military aircraft in a radar display are classified according to threat. The task is divided into four sub-tasks: target selection, information retrieval, score calculations and feedback processing. The authors were interested in describing the microstrategies used by participants in this microworld. The authors identified different strategies for the different sub-tasks involved. For example, for the *target hooking* subtask they identified a strategy called *Look-after* where the participant encodes the target number, clicks on the target's icon, looks at the information window (this windows contains information such as the speed and bearing of the target), and make a verification by matching the encoded number with the number displayed in the information window. Another found for the same subtask was called the *Look-before* strategy, in this strategy the cursor is placed over the intended target, attention is switched to the information window, and a click in the icon is executed whereby the participant can see how the information window changes at that moment. During the cognitive modelling phase the authors created different microstrategies for the various subtasks involved: target selection, information retrieval (either apply the Look-after or the Look-before strategy), score calculation (to assess the threat level of the aircraft) and feedback processing. A model is run using a combination of microstrategies and patterns of similarity are determined by comparing the output of the model with that of participants, in the end the researchers selected the combination of microstrategies that best fits each participant. This research work shows how microstrategies can be identified by combining data analysis and cognitive modelling and provides an important level of detail in explaining that performance.

Veksler, Gray, & Schoelles (2007) created a cognitive model for a *Table Decision* task in which information is obtained by clicking cells in a 6 x 6 matrix (see section 2.2.3.4). This task is static in nature thus allowing a considerable time for making decisions. The authors identified three subtasks: uncover a cell to reveal its value (using a mouse click), add the value of the current cell to the total for each alternative, and (because the goal is to find the aircraft with the highest threat) update the alternative with the highest value. They created four models following an incremental approach: when a new model was created it inherited the properties of the previous model. The found that the last model, the one with the highest complexity, shows the best fit to participant data; this model interleaves cognitive with perceptual-motor operations, prepares motor actions, and alternates transitions between rows, that is: after disclosing all cells from left to right in a single row, it starts disclosing the next row of cells from right to left reducing the length of mouse movements.

Byrne & Kirlik (2005) use a task that involves moving an aircraft to the correct gate in an airport after landing. This study is focused on decisions related to the selection of paths to arrive at the target gate. Five strategies with different levels of speed and accuracy were identified in this study which considered different scenarios (i.e. airports). A fast but inaccurate strategy is to remember the required turns (there are many gates and airports in this task). A second strategy is to make turns towards the gate, and a third one is to select the turn that reduces the distance between the aircraft and the gate the most (the authors consider this strategy as more effective than strategy 2). A fourth strategy is the slowest because spatial knowledge is used, but it is only accurate if the pilot has experience of a specific airport. The last strategy is to choose the turn randomly. Aided by an ACT-R cognitive model, they found that participants use the most accurate strategy possible given the time available; specifically, when time is short, the problem solvers tend to rely on their past experience of the task (i.e. typical taxi routes), because it is computationally cheaper.

Lovett & Schunn (1999) propose a model called RCCL (pronounced ReCyCle). Within this model a strategy choice is a function of how base-rate and case-specific cues are combined. Base rate refers to how often a strategy is successful and the information about the current situation can be used for predicting the success of an alternative. The RCCL model has four stages: represent the task, construct a set of strategies, choose among them according to their success rates and learn new success rates based on experience. In RCCL a task representation includes a subset of all possible task features; this selection is based on background knowledge and the salience of different features. In the example given by the authors, a driver represents time of day and the best route to take. This information provides a knowledge base for taking a decision. Strategies are expressed as rules with conditions and actions, for example: if the time of day is morning, then take the highway route. As mentioned, each strategy's estimated success rate is learned via experience. For the purpose of being adaptive, the stage in which success rates are learned is critical. The most important predictions of this model are that people will learn to prefer the strategies that have higher base rates of success and that more strategy change will occur in tasks that have met with low success rates. In the study of Cañas et al. (2005) more strategy variability is found in the variable training programme where trials of low complexity and high complexity are mixed and there are more task features involved (e.g. the wind changes intensity during trials) compared to the constant training programme where trials always follow the same course. The RCCL model predicts more strategy change in the variable training condition due to low success rates being the product of more complex trials plus the addition of the new feature related to wind strength change.

The work of Payne et al. (1993) is relevant for the cognitive modelling of strategy choice. These authors consider that production rules can be used to transform knowledge states in order to execute decision strategies. A strategy has a range of characteristics that are taken into account when making a decision: availability, accessibility, ease of execution and perceived benefits (Payne et al., 1993). Using as an example the Cañas et al. (2005) study (section 4.1), availability varies for the different training programmes where the different distribution of terrain and wind conditions facilitate or hinder the applicability of strategies. Similarly, the condition where the environment is exactly the same for all sixteen trials facilitates accessibility to a strategy implemented in the previous trial versus the case where the problem solver is presented with a different scenario on each of the sixteen trials where the changing conditions may hinder the accessibility of a previous strategy. Processability and perceived benefits are further explained in the following chapter (section 4.3.2) where a detailed analysis of FireChief strategies is provided. For Payne et al. (1993) the question is how one decides how to decide. The authors offer a definition for a decision strategy as a sequence of “mental and effector operations used to transform an initial state of knowledge into a final goal state of knowledge where the decision maker views the particular decision problem as solved” (p. 8). In the context of this research what is relevant in the research of Payne et al. is the idea that individuals try to find strategies that will yield high degrees of accuracy (in relation to performance) for reasonable amounts of effort. So, if the level of accuracy of two strategies is similar, the problem solver will use that which involves as little effort as necessary to solve a problem and, if the level of desired effort is equivalent, the problem solver will try to choose the strategy that is the most accurate. Therefore, there is a compromise between the desire to make the most correct decision and the desire to minimize effort. Another observation by the same researchers is that the levels of accuracy and effort for a given strategy change depending on task characteristics. According to Gerjets et al. (2000) the adaptive selection of strategies should be of major importance for success in learning and problem solving. In the study of Charman & Howes (2002) practice resulted in more efficient performance on the task due an adaptive selection of problem solving strategies.

Different strategies are characterized by different levels of accuracy and required effort but, in order to apply these ideas, there must be a way of measuring effort and accuracy. According to Payne et al. (1993) an effortful decision process is identified by a high number of operators or more demanding operators. Newell & Simon (1972) have suggested that cognitive processing effort can be measured in terms of the number of Elementary Information Processes (EIPs) needed to complete the task, while Shugan (1980) suggests that decomposing decision strategies into components can provide estimates of their relative costs; a similar proposal is to decompose strategies into smaller components called microstrategies (Schoelles & Gray, 2000). Payne, Bettman, and Johnson (1993) used Elementary Information Processes to calculate the mental effort of each decision strategy in their choice task, such as reading an item of information, comparing two items of information, multiplying or adding items of information, and eliminating items of information. They present a framework for measuring cognitive effort and accuracy for choice strategies. The relation between ACT-R production rules and EIPs is straightforward: Payne et al. argue that EIPs could be used as the actions of productions. On the other hand accuracy can be measured using a function such as maximization of expected value (EV). The task used by Payne et al. (1988) is a choice problem where participants search among probabilities and the values associated with them to compare the different decision strategies. Ten decision strategies were investigated in this research but only two of them are presented here, with the aim of illustrating how EIPs are used for comparing strategies. The first strategy is called weighted additive (WADD), it evaluates every available option by considering all relevant attributes and the option with the highest evaluation is chosen; this is an information-intensive strategy. The second strategy is called satisficing (SAT). In this strategy options are evaluated one at a time by comparing each of their attributes with a cut-off value. An option is rejected if any of its attributes is below the threshold and the first alternative that passes the cut-off is chosen (there is no guarantee that the best choice is selected). The WADD strategy is a compensatory strategy whilst the SAT strategy is a non-compensatory one (see section 3.4). On average, the cognitive effort required to execute the WADD strategy is greater than the SAT strategy. This effort is also related to the complexity of the problem: the number of EIPs required for the WADD strategy is 28 for a trial with two options and two attributes whereas 400 EIPs are required for a trial with eight options and eight-attributes. In their study it was found that the SAT strategy approximated the accuracy of the WADD strategy using a fraction of the effort and that time pressure often truncated the execution of the WADD strategy and hence its accuracy was reduced.

Taatgen (2005) observed that complex dynamic tasks involve multitasking. A model was developed for the CMU-ASP task. In this task participants classify aircraft by asking information of the system. The most important information for classifying aircraft is speed and altitude. Aircraft are classified by means of the 12 function keys on the keyboard (F1 to F12). Each one of these 12 keys has a label that describes their function and that change dynamically depending on the context. For instance, F6 can activate the *Track Manager* but after the *Track Manager* is activated the label associated with the F6 key will change. Because the role of these function keys changes throughout the simulation the authors argue that considerable learning is necessary to achieve proficiency in this aspect of the task: instead of spending time reading the labels associated to the function keys participants can learn the value of these labels and therefore complete the task faster. To execute this task three stages are necessary: select a plane, determine its identity, and enter its identity into the system. Points are awarded for successfully classifying aircraft. An important feature of the CMU-ASP model is that it needs to keep track of where it is in the problem solving process, a feature that exerts demands on WM. Taatgen (2005) developed this model following a design in which a weak control structure with as few control states as possible were sought. Taatgen's model is able to interpret a set of instructions and learn new rules through production compilation (see section 2.4.1.5). In this model instructions are retrieved by paying attention to cues. Its instructions are organized into unit tasks that reflect the three stages needed to classify aircraft found during task analysis. Only one set of instructions is active at a time, that is, only the set of ACT-R rules that belongs to the active unit set can be fired. This feature reduces the number of different chunks that can be placed in the goal buffer, therefore reducing the total amount of possible goal states (which is the goal of the *minimal control principle*). The CMU-ASP task is fairly dynamic and the model is able to cope with it by having its rules organized into sets and by being sensible to the cues received from the environment.

#### 2.5.2.1. Identifying constraints in strategy use

An important endeavour that should be taken into account when modelling behaviour is the definition of how constraints, on the way a sequence of actions implement a strategy, are applied. A coherent set of constraints on participants' behaviour can be extracted by considering the demands imposed by complex dynamic microworlds described in this chapter. This set of constraints is combined with those obtained from the analysis of data described in the next chapter. Different ways of representing constraints have been proposed in the cognitive modelling research community. Some approaches use formal languages for specifying constraints. Sceptic (Cooper et al., 1996) is an executable specification language implemented in Prolog. In Sceptic, data is represented as a relation between zero or more parameters. Sceptic maintains a system's state (defined at the beginning of the simulation) which is modified as execution progresses. During execution procedural knowledge can modify this state and trigger more rules (each Sceptic rule has 3 elements: a trigger, conditions and actions). The modelling methodology promoted by Sceptic is based on a distinction between theory and implementation (Cooper et al., 1996). The theory level specifies a systematic methodology for theory articulation. The implementation level refers to the mechanisms that have to be implemented to make the program executable in order to test the core assumptions of the theory. Howes, Vera, Lewis, & McCurdy (2004) propose a similar approach called Cognitive Constraint Modelling (CCM). In CCM, behaviours are described via constraint satisfaction over architectural, task and strategy constraints. Constraints are specified in terms of predicate calculus statements relating entities in the environment, task and psychological processes. Entities are represented as a set of attribute-value pairs. CCM depends on optimisation and cascaded information-processes. Cascading requires the specification of the processing resources for communication between processes. The authors give as an example the combination of two preparations and two executions of a click command to show the limits of temporal dependencies. The sequence init(x), init(y), click(y), click(x) does not violate a temporal dependency, but does not respect the ACT-R architectural constraints: that a buffer can only store a single chunk, so, in this case, the second preparation would reset the first one cancelling the execution of click(x). In ACT-R, production rules are in charge of cascading information-processes.

To illustrate the use of CCM, a FireChief constraint expressed in CCM is presented next. The constraint is described as follows: "A Control Fire command starts after a C key is pressed and the mouse pointer is located above a free truck. The duration of the Control Fire is of two seconds."

$$
\begin{aligned}
&\forall Pi : \{(\text{isa, process}), (\text{name, ``press-key''}), (\text{pressed-key-is, ``C''}),\\
&(\text{start,Si}),\ (\text{duration, Di})\} \subset\ Pi\\
&\wedge \text{Ci}\,\{(\text{isa, condition})\ (\text{name, mousePointerOver}), (\text{unit, truck})\}\\
&\wedge \text{Cj}\,(\text{isa, condition}), (\text{name, StatusUnit}), (\text{status,Free})\} \Rightarrow\\
&\exists\,\text{SPk} : \{(\text{isa, Simulation Process}), (\text{name, ControlFire}), (\text{start,Sk}), (\text{duration, 2 s.})\}\\
&\wedge\ \text{Si} + \text{Di} = \text{Sk}
\end{aligned}
$$

In CCM, after all constraints are defined, a constraint satisfaction engine can be used to get a sequence of valid actions that reports the greatest utility according to an objective function such as minimizing time, so the model's behaviour is controlled by objective functions. Ohlsson (2007) uses an approach similar to CCM: the state constraint representation. This representation applies to declarative knowledge that is represented as a pair of patterns: the relevance criterion and the satisfaction criterion. These constraints are used for guiding the search for solutions using the Rete algorithm. Ohlsson affirms that a constraint base supports judgement by offering constraint functions that identify search states consistent with the principles of the domain. In the FireChief model constraints are embedded in the structure of the task, the ACT-R architecture and the production rules. Productions rules are fired if their conditions are met and these conditions are a combination of ACT-R buffer contents. Sequential dependencies are implemented by manipulating the contents of the goal buffer. If rule B must fire after rule A, the latter will modify the goal buffer in order to satisfy a condition of rule B. Because the model interacts with the simulation in the same fashion as human participants, many constraints are imposed by the simulation. For instance, if the simulation detects that a C key is pressed when the mouse pointer is located over a truck, a *Control Fire* will be started in that cell and will last 2 seconds. This is the framework within which FireChief strategies are implemented.

## 2.6. Summary

This chapter described how the study of CPS can be undertaken from the perspective of the interaction of the problem solver, the task and the environment (Jonassen, 2000; Frensch & Funke, 1995). From the set of CPS performance determinants described in section 2.2.3 the discussion was centred on strategy selection and implementation. A particular set of tasks called microworlds were described, examples of representative tasks were provided and their key characteristics were presented before discussing the cognitive demands imposed by these tasks. The importance of using strategies is considered in this section from the standpoint of dealing with microworlds. Cognitive modelling is introduced as an approach for studying CPS behaviour and section 2.4.1 describes how several characteristics of the cognitive architecture ACT-R make it suitable for creating a cognitive model of FireChief behaviour, mainly its ability to continuously adapt its behaviour by processing environmental feedback. The next chapter describes the process of analysing the data extracted in the Cañas et al. (2004) study in which the detailed interactions of participants with FireChief are taken into account to identify a comprehensive set of strategies.

# 3 Data Analysis of CPS behaviour

The data to be modelled originated from the study of Cañas et al. (2005) using the FireChief microworld where various topics related to strategy use were explored. A description of this dataset and several interesting interactions comprise the first section of the chapter. Results obtained by Cañas et al. are taken into account in this research, but the goal of implementing a cognitive model demanded a more detailed level of data analysis. The chapter continues with a description of brand new data analysis process that resulted in the definition of a hierarchy of strategies. A set of metrics used to measure and discriminate these strategies are also included in this chapter.

## 3.1 The Cañas et al. (2005) Dataset

In the Cañas et al. (2005) study each participant interacted with 24 FireChief scenarios. The first 16 scenarios were considered as the training phase and the last 8 scenarios as the testing phase. There were two different training programmes: constant and variable. In the Constant Training (CT) the environment is exactly the same for all sixteen trials. In the Variable Training (VT) the problem solver is presented with a different scenario on each of the sixteen trials. There are also two testing conditions: either the direction in which the wind blows is changed (WC) or the effectiveness of units in extinguishing fires is reduced (ER). Table 3.1 shows the experimental groups of the study, the number between parentheses represents the number of members in each group. In table 3.1 group CTW represents the group which received CT and was tested with WC; the VTE group received VT and was tested with ER, and so on; this notation will be used throughout the remaining chapters for referring to the different groups.

| | **Wind Change** | **Efficiency Change** |
|---|---|---|
| **Constant Training** | CTW(18) | CTE (20) |
| **Variable Training** | VTW (16) | VTE (18) |

**Table 3.1: Experimental groups of the Cañas et al. (2005) study**

Cognitive flexibility “is the human ability to adapt the cognitive processing strategies to face new and unexpected conditions of the environment” (Cañas et al., 2005, p. 2). Cañas et al. (2005) hypothesized that the type of training would have an impact on the level of cognitive inflexibility of individuals: people in the VT group would show greater facility for changing strategies than individuals trained repeatedly on the same (constant) scenario. The study found that the type of training did affect cognitive inflexibility in this way. The theoretical explanations for this phenomenon come from studies about skill learning (Ackerman 1992; 1988; Rasmussen, 1983). The explanation is that participants in the CT group have the opportunity to consolidate their strategies and hence generate quick, fluid actions; while people in the VT group keep executing more controlled, but flexible, actions. When the testing phase takes place, people in the CT group are less (cognitively) flexible in adapting to the new demands of the task. With regard to strategy learning, participants in the CT group have the opportunity to practise the same strategy over and over again, and more practice yields more automatic behaviour. The authors also found that the changes introduced in the environment during testing generated significant effects on performance and that these changes affected the participants differentially depending on the strategy that they were putting into practice. It was found that strategies which rely on the execution of Control Fire (CF) commands are more affected by changes in the direction of the wind. The explanation given by the authors was that "the place where the fire control should be located depends on where the fire spreads towards" (Cañas et al., 2005, p. 4) and when the wind changes direction it is harder to locate a suitable position for a CF command. It was also observed that strategies based on Drop Water (DW) commands are more affected by the reduction in the efficiency of appliances. The Cañas et al. study was limited to finding interactions between experimental conditions but no attempt to explain these interactions, beyond theoretical explanations, was pursued. The following sections describe in detail this dataset which, considering the research goal of studying strategy use in complex dynamic tasks, provides a rich set of combinations of task manipulations and interactions, a considerable amount of empirical data to perform a fine-grained analysis and also represents a challenging task for modelling.

### 3.1.1 Training programs

Remember that the instruction given to the participants in both groups is to save the landscape from destruction by fire so as to get as high a score as possible. In the context of the CPS paradigm described in chapter 2 the different training programmes can be seen as different kinds of obstacles that the problem solver needs to overcome using his or her limited cognitive resources. A FireChief scenario is specified by various properties: the landscape distribution, the number, position and type of initial fires, the number and direction of wind strength changes and the initial position of units. A single trial lasts 260 seconds and the simulation is refreshed each 200 milliseconds. Within FireChief each block of 200 milliseconds is called a *generation*, therefore each trial lasts 1300 generations.

#### 3.1.1.1 Constant Training (CT)

As can be seen in table 3.1 groups CTW and CTE share the same training programme. Under this programme participants interact with the same task configuration (the trial "C") 16 times.

##### Trial "C"

In this trial wind strength remains at six throughout and the wind always blows to the east. There are two blocks of fire close to each other. Assuming no intervention both initial fires spread fairly quickly to the east. At second 169 the fire arrives at the eastern limit of the scenario. Only the three top lines of landscape remain undestroyed. The remaining fire spreads north and south but at a slower pace. By observing protocols it was determined that the critical time for the CT scenario is around second 70. By this point in time, if the fire is not under control (there are many ways of achieving this) there will be a considerable amount of landscape burning and it will be difficult to stop it. Remember that after a cell is destroyed fire will expand to the neighbouring cells; if this phenomenon is occurring at all times for many cells the complexity of the trial could reach a level beyond the participant's capacity to extinguish the fire. Because this trial is repeated sixteen times during the CT programme participants have the opportunity of becoming familiar with this trial. In this trial participants may learn several aspect of the FireChief task but mainly that high wind strength produces strong fires. The first row in table 3.2 shows the characteristics of this trial.

### 3.1.1.2 Variable Training (VT)

As can be seen in table 3.1 groups VTW and VTE share the same training programme. In the VT programme every one of its 16 trials is different. Variations among trials include the diverse landscape composition, altered initial position of units and fires, but more importantly, there are changes in wind conditions. As an example, in trial number 3 there is one mature fire at the beginning of the trial and the initial wind strength is three (low intensity), after 40 seconds a spot fire appears then 110 seconds later wind strength increases to seven. In contrast, in trial number 13 there are two initial mature fires and the wind strength is seven (high intensity) after 140 seconds the wind strength decreases to four and then there is another change in wind strength after a further 80 seconds where the wind strength increases to six. To give an idea of the comparative damage produced by the fire in these two trials the base performance (explained below) for trial number 3 is 62.49 and for number 13 is 37.15.

| Trial Number | CPX | Terrain Value | # Initial Fires | | New Fires (gen) | | Initial Wind Str | Wind Str Shifts (wind str/gen) | | Base Perf | Avg. Perf |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mature | Spot | Mature | Spot | | | | | |
| C | H | 1702 | 2 | 0 | 0 | 0 | 6 | | | 50.6 | 69.9 |
| 1 | VH | 1808 | 2 | 0 | 0 | 0 | 7 | | | 43.4 | 43.8 |
| 2 | M | 1708 | 2 | 0 | 750 | 0 | 4 | 8/1000 | | 41.8 | 76.5 |
| 3 | VE | 1704 | 1 | 0 | 0 | 200 | 3 | 7/750 | | 62.5 | 93.5 |
| 4 | H | 1630 | 1 | 1 | 0 | 0 | 6 | 7/750 | 4/1000 | 56.4 | 60.8 |
| 5 | VE | 1688 | 0 | 3 | 0 | 0 | 4 | 7/750 | | 47.2 | 93.2 |
| 6 | VE | 1814 | 2 | 0 | 0 | 0 | 4 | 3/750 | 5/900 | 79.3 | 96.7 |
| 7 | VH | 1726 | 2 | 0 | 0 | 1000 | 7 | 6/500 | 2/1000 | 51.2 | 54.8 |
| 8 | VE | 1740 | 1 | 1 | 0 | 850 | 3 | 7/1000 | | 86.8 | 99.2 |
| 9 | H | 1688 | 2 | 0 | 0 | 900 | 6 | 4/800 | | 66.3 | 72.5 |
| 10 | VE | 1824 | 0 | 3 | 1100 | 0 | 3 | 7/750 | | 56.5 | 94.0 |
| 11 | H | 1670 | 1 | 1 | 700 | 0 | 7 | 3/500 | 7/1000 | 54.3 | 70.0 |
| 12 | VE | 1762 | 1 | 1 | 0 | 750 | 3 | 7/900 | | 70.1 | 98.1 |
| 13 | VH | 1714 | 2 | 0 | 0 | 0 | 7 | 4/700 | 6/1100 | 37.2 | 48.0 |
| 14 | VE | 1726 | 1 | 2 | 0 | 0 | 4 | 7/600 | 3/900 | 66.9 | 93.0 |
| 15 | VH | 1802 | 0 | 3 | 0 | 0 | 7 | 5/700 | | 33.4 | 43.5 |
| 16 | H | 1666 | 2 | 0 | 0 | 0 | 5 | 3/700 | 7/900 | 55.8 | 81.6 |

**Key:**
CPX = complexity
gen = generation
Str = strength
Perf = performance

**Table 3.2: Constant (C) and Variable Training (1-16) Trials**

The configuration of each trial is presented on table 3.2. The second column in this table shows the complexity (CPX) of the trial that can be VE (very easy), E (easy), M (medium), H (hard) and VH (very hard). This complexity was obtained by applying a range of criteria: the average performance of participants, the wind strength and the number of fires. Value refers to the value of the terrain; the value of a terrain is a function of its constitutive cells. The value of a landscape impacts performance because a more valuable landscape represents a greater opportunity for losing points. The following columns in table 3.2 describe the number of mature and spot fires both in the initial scenario and newly igniting during the trial. For fires igniting during the trial the number indicates the generation (remember that a generation is equal to 0.2 seconds). The next column describes the initial wind strength and subsequent shifts of wind strength. The penultimate column shows the base performance (the trial is run letting the landscape burn without intervention and the final performance score was recorded). In the last column the average performance of participants during the training phase is presented.

### Trial 1

In this trial there are two blocks of fire close to each other and wind strength is 7. Both initial fires spread fairly quickly to the east. After 105 seconds the first fire arrives at the eastern limit of the scenario. The rest of the time the remaining fire spreads north and south. The problem solver needs to issue a high number of commands at the beginning of the trial in order to gain control of the fire quickly; otherwise the fire front grows quickly. One aspect that should be learned by the problem solver is that the fire spreads in the wind direction. The problem solver may also discover that some fires, (the ones with higher strength, their icons depicting a bigger fire) cannot be extinguished by copters or trucks dropping water.

### Trial 2

There are two initial fires wide apart, but the wind strength is low (4) for many seconds, hence the fire spreads at a very slow rate. When a mature fire appears later the wind strength remains low so it does not represent a major threat. But at generation 1000 a drastic increment in wind strength to 8 makes the fire spread considerably faster and in the last 400 generations the fire destroys a lot of terrain. In this scenario participants experience for the first time a shift of wind strength and the spontaneous ignition of new fires.

### Trial 3

At the beginning there is one mature fire in this scenario, but a spot fire appears after 40 seconds. The initial wind strength is 3. During the time that the wind strength is 3, the initial fire has minimal development, the same is true for the spot fire. After 150 seconds (750 generations) the wind strength increases and the fire spreads quickly eastwards. At the end of this trial the problem solver may learn that a second kind of fire can appear (the spot fire). This kind of fire is less aggressive than a mature one but it is harder to locate within the landscape. The problem solver may also learn that very low wind strength reduces fire development.

### Trial 4

A single mature fire grows at a steady pace followed by the onset of a spot fire. The spot fire and the mature fire become one at generation 375 and the mature fire reaches the eastern limit of the landscape by generation 590. When the first wind strength shift occurs, increasing to 7 (generation 750-1000), the fire has spread eastwards and reaches the landscape limit. Only four lines in the north are not destroyed. The second shift of wind (generation 1000), where wind strength drops to 4, leaves a considerable number of low intensity fires burning until the end of the trial. The problem solver has experienced that spot fires can appear and that wind strength can reduce.

### Trial 5

This is the first trial in which there are no initial mature fires. A combination of spot fires and low initial wind strength (4) makes this trial very easy. Perfect performance of 100% is easy to obtain. The three initial spot fires do not develop during the first part of the trial. The rate of spreading is very low until the wind strength increases to 7 (generation 750) making the fire spread quickly to the west consuming almost every cell in the landscape except the north-west corner. The problem solver has now experienced that mature fires are not always present in trials.

### Trial 6

The two mature fires are far apart from each other. Both fires start at the same X-coordinate but are separated by 10 cells in the Y axis. The wind strength is low (4) deterring the development of the fires. When the wind strength reduces to only 3, the fire development almost stops. However, when the second shift occurs, increasing to 5 (generation 900) the western fire is capable of catching up with the eastern one by generation 1200 although in general the fire is not aggressive.

### Trial 7

The two mature fires are wide apart. The eastern fire is strongest at the beginning of the trial and by generation 175 it reaches the eastern limit of the landscape. The western fire catches up with the eastern fire in generation 850. The first shift in the wind strength from 7 to 6 (generation 500) reduces the spreading rate but, by this time, the eastern fire has destroyed almost all existing terrain. The second shift to 2 (generation 1000) almost freezes the fire's development leaving a considerable number of low strength fires. A salient aspect of this trial is that one of the fires quickly arrives at one edge of the landscape; it is possible that problem solvers are now more aware of the consequences of certain combinations of fire position and wind direction.

### Trial 8

This trial is similar to trial 3. For 1100 generations the wind strength is 3 so the one mature fire shows almost no development, and the one spot fire shows none. A second spot fire appears at generation 850. When the wind strength increases to 7 (generation 1000) the mature eastern fire quickly reaches the eastern border and the two spot fires merge into one and continue their way eastwards. The role of wind strength as a determinant of fire spreading rate may be reinforced in this scenario.

### Trial 9

Two mature fires are wide apart. The eastern fire is close to the eastern limit of the landscape. The setup is similar to trial 7. Both fires have the same intensity. At generation 235 the eastern fire reaches the eastern limit. The western fire never catches up with the eastern one. When the shift of wind strength from 6 to 4 occurs (generation 800) the western fire has not reached the eastern limit and, although there is some fire development, it never does. A spot fire appears at generation 900 when the wind strength is low and so does not develop. The problem solver can make use of the limits of the landscape to his or her advantage as they represent natural barriers to the fire.

### Trial 10

This can be a very easy trial as the spot fires can be attended to quickly. The three spot fires do not develop until generation 525. When the strength of the wind increases from 3 to 7 the eastern fires merge into one and soon reach the eastern limit. The western fire also soon catches up with the other fires (generation 900).

### Trial 11

At the beginning of the trial there is one mature fire in the west and a spot fire nearby. At generation 700 another mature fire appears in the south-east. The mature fire located in the west starts developing from the beginning of the trial; the spot fire destroys its containing cell 30 seconds after the beginning of the trial. The wind strength is high (7) so the spot fire reaches the eastern limit 5 seconds later. At generation 490 the mature fire located in the west catches up with the spot fire located at the eastern limit. When the wind strength decreases to 3 (generation 500) most parts of the terrain to the east of the initial fire are destroyed. While the wind strength remains low there is almost no further fire development. A second shift of wind strength from 3 to 7 (generation 1000) accelerates the development of the mature fire that appeared in the south and which quickly reaches the eastern limit of the landscape. In this trial the shifts of wind strength are large and represent major differences concerning fire behaviour. Particularly in this trial, it is possible to attack the spot fire directly even though the wind strength is high; this phenomenon gives more hints to the problem solver about the behaviour of different strengths of fire.

### Trial 12

Both the one mature fire and the one spot fire develop a little while the wind strength is 3. This situation remains the same for three quarters of the trial. When the wind strength shift to 7 occurs (generation 900) the spot fire expands and reaches the eastern limit at generation 1125 while the mature fire catches up with the spot fire at generation 1185. The spot fire that appears at generation 750 is soon engulfed by the rapidly spreading mature fire. This trial is similar to trial 8. The role of wind strength as a determinant of the fire spreading rate is reinforced. Considering the experience of similar trials, it is most likely that the problem solver is by now more aware of the importance of acting quickly while wind conditions are favourable.

### Trial 13

The two initial mature fires are close to each other; this trial is similar to trial 1. Both fires are initially located in the west of the landscape. The wind strength is high (7) so quickly both fires merge into one and advance eastwards very fast. By generation 590 (118 seconds) the fire arrives at the eastern limit and by the time of the first wind strength shift to 4 at generation

700 the fire has consumed almost all the possible landscape. The second shift increases wind strength to 6 (generation 1100). The rapidly spreading fire front has a very high intensity and it is not possible to apply DW commands effectively.

### Trial 14

One mature and two spot fires are wide apart. The initial strength of the wind at 4 allows for a certain amount of development of the mature fire, and one of the spot fires soon reaches intensity beyond the reach of the DW command. The other spot fire develops more slowly. Although an increment in wind strength to 7 (generation 600) only lasts 300 generations, it is enough for all the fires to merge into one and reach the eastern limit at generation 840. When the second shift in wind strength to 3 occurs at generation 900 there are still some regions that can be saved from the fire. Again, if the problem solver has learned from past experience, they are prompted to attack the fire quickly, as in past trials initial low wind strength is followed by an aggressive increment.

### Trial 15

The three spot fires are wide apart forming a triangle. The high wind strength of 7 that does not change until generation 700 produces fast development of the fire. The fire located in the north merges with the one located in the east at generation 275 and reaches the eastern limit of the landscape at generation 475. The southern fire also reaches the eastern limit at generation 680 burning the remaining cells to the south. By the time the wind drops its intensity to 5 at generation 700 most parts of the terrain are destroyed. This is one of the hardest trials.

### Trial 16

In this trial there are two fires in the north. The uppermost is located closer to the eastern limit. Both fires are equally strong. The uppermost fire arrives at the eastern edge at generation 950 when the wind strength is 7. Both fires merge eventually and destroy everything to the east of their initial starting point. The distribution of fires in this trial is very similar to that in the CT scenario but the wind strength is lower.

## 3.1.2 Comparison of training programmess

As mentioned, Cañas et al. (2005) hypothesised that participants would consolidate their strategies in the CT trial. A common assumption is that people become effective when presented with recurring problem patterns (Anderson, 1983) so it is expected that members of the CT group will arrive at a good level of performance over successive trials and performance data confirms this hypothesis. In the end repeated exposure to a problem produces a certain degree of expertise and should therefore lead to better problem performance and representation. Another characteristic of FireChief that promotes performance improvement in the CT condition is its high level of consistency. According to Ackerman (1993), when the task is consistent, performance improvements result with increased levels of task practice. Also Rasmussen (1983) stresses the importance of the existence of constraints between human actions and their effects for promoting meaningful interaction with the task. Given that the different FireChief commands produce the same results under the same conditions the problem solver can generate an adequate representation of the task. It is expected that after the problem solver has found a suitable strategy for the CT trial some degree of automaticity can be achieved. According to Gonzalez et al. (2004) decisions became increasingly similar with task practice as a consequence of a better understanding of the relevant variables of the system, so it is expected that strategy variation will decrease as more trials are completed. Learning the behaviour of FireChief under repeated conditions should improve the process of choosing operators (commands) because there will be more chance of knowing how to better implement the operators, to assess their applicability and effectiveness, although there would be less exploration of FireChief (this issue is explored using the cognitive model). Cañas et al (2005) argue that participants in the CT group assign their resources to the continuous practice of a strategy that has proven successful in the CT trial (which is repeated over and over). They expected that participants in this group would perform well and show more cognitive inflexibility during testing.

This richer diversity of problems allows participants in the VT group to gain experience about a range of strategies. In the case of the VT programme different approaches for fighting the fire in a particular trial may be tried until an adequate sequence of commands is found. Following ideas from Thunholm (2005) the difference between training programmes may result from the fact that participants in the VT group perform a situation assessment process (see section 4.3.1) while participants on the CT programme skip this step.

Training programmes become particularly relevant when a change in the environment is introduced in the testing phase. The question is whether the different training programmes generate different ways of dealing with environmental change. An intuitive approach used for distinguishing the training programmes is the novice-expert paradigm. Participants in the CT are in some ways experts when dealing with the CT scenario. The first test trial looks identical to all the trials in the CT so will be familiar to the CT group, but for participants in the VT group the first test trial will look like a trial with yet another new and different configuration. The distribution of landscape and units is the same as for the trials in the CT programme, making participants in the CT group experts in comparison to participants in the VT group. For both groups the consequences of the changes in the system configuration for testing are not immediately visible. In the condition where the wind changes direction, the first change in wind direction occurs at generation 600, so for the first 600 generations participants in the CT programme presumably think that they are solving the same trial as in the previous sixteen occasions. In the condition where the efficiency of appliances is reduced, in order to notice that copters and trucks are less able to extinguish fires it is necessary to execute a DW command over a fire of certain strength to discover that instead of the command succeeding an alarm will sound.

From a different point of view (Payne et al., 1993), differences between the training programmes can be traced to differences in how context variables are processed. In FireChief the process variable “wind conditions” has a significant impact on strategy selection. Because the CT programme does not present shifts in wind strength presumably participants in this group do not continuously check this value on the display. On the other hand participants in the VT group may be more prone towards checking this value due to the considerable amount of wind strength shifts. In the latter case the variable wind conditions are processed continuously while in the former this variable may be seldom checked. This difference may have an important consequence for the testing condition where wind changes direction in the trials for both groups: participants in the VT programme may detect a change in the wind direction earlier in the trial just because they are more alert to changing wind conditions. The uncertainty about the environment requires the participant to modify his or her strategy continuously forcing them to attend to any change that could take place when a trial is presented. In the end it is expected that participants will be more attentive and less prone to cognitive inflexibility.

### 3.1.3 Testing phase

From the seventeenth trial until the end of the experimental session participants must deal with one of two types of environmental change. The landscape distribution, the number of fires and the initial location of units is the same as in the C trial for all eight testing trials.

#### 3.1.3.1 Wind Direction Change

In the Wind Direction Change (WC) condition the wind starts blowing towards the east and after 300 generations the wind switches to the NW and thereafter shifts 60 degrees anticlockwise every 300 generations, finally ending once again in the east. These shifts in wind direction have a dramatic impact on fire development. Assuming no intervention the fire destroys almost all the terrain as the most eastern section of the landscape can now be destroyed when the wind blows eastwards.

#### 3.1.3.2 Efficiency Reduction

The second type of change is called Efficiency Reduction (ER). The efficiency of an appliance refers to its ability to extinguish fires. A reduction in efficiency means that units are less able to extinguish fires. During the testing phase efficiency is halved so the ability of the unit to extinguish strong fires is seriously affected. In practical terms, during testing units are able to extinguish fewer fires than during training. Assuming no intervention the terrain suffers the same destruction as the C trial.

#### 3.1.3.3 A comparison between testing conditions

The nature of the changes under the two testing conditions is quite different. A change in wind direction is more salient: there is a specific icon within the graphical display that can be queried to find out the wind direction and fire behaviour is noticeably different. A reduction in the efficiency of the appliances is only noticeable when a participant tries to issue a DW command on top of a strong fire and instead of starting to drop water the simulation emits an alarm. Participants in the VT group have more experience in dealing with this kind of alarm due to the fact that they have faced trials with stronger wind and therefore more intense fires before.

When wind changes direction, if participants are adaptive, there should be an abandonment of the use of Control Fire commands (Cañas et al., 2005). In the testing phase participants in the VT group will detect that a change in the environment occurs but that the same trial is being repeated over and over. The feedback they receive in the first testing trial should result in the quick selection of a more appropriate strategy. Participants in the CT group should take more time before switching strategies. But if the right strategy is chosen in trial 17 both groups should repeat this strategy choice if it receives good performance feedback, because the environmental conditions are the same. In fact, cognitive inflexibility is exactly what was observed in the Cañas et al. (2005) study for the Constant Training group dataset. This inflexibility is described as a tendency to maintain the same strategy even when changes in the environment make it inappropriate.

Continuing with the discussion about the importance of "*gaining control*" started in section 3.2.4, the precise form of the fire-fighting process depends on the strategy selected but, in all situations, some area will be covered by this process at a speed that will be a function of the number of units used and the kind of commands executed. If the problem solver makes use of all four available units, the amount of area covered may be bigger and, because a *Control Fire* command takes less time to complete compared to a *Drop Water* command, using a mixture of commands may increase the total number of commands issued. Figure 3.1 shows a second spatio-temporal diagram representing a VT scenario. In the VT programme there could be changes in wind strength. These changes noticeably alter the rate at which the fire spreads. In the first stage depicted in figure 3.1 the problem solver is able to control the fire by issuing *Drop Water* commands but after a period of time the wind conditions change and the rate at which the fire spreads increases. At this point, the sensible thing to do is to issue *Control Fire* commands along with *Drop Water* commands (the *Control Fire* commands are aimed at halting fire development). In the last phase wind conditions change again, allowing the participant to regain control over the fire. As participants try to gain control over the complex situation in front of them, they generate a rich set of strategies which are the focus of this research. As discussed in the previous section, the use of strategies appears to be an effective way to deal with complexity.

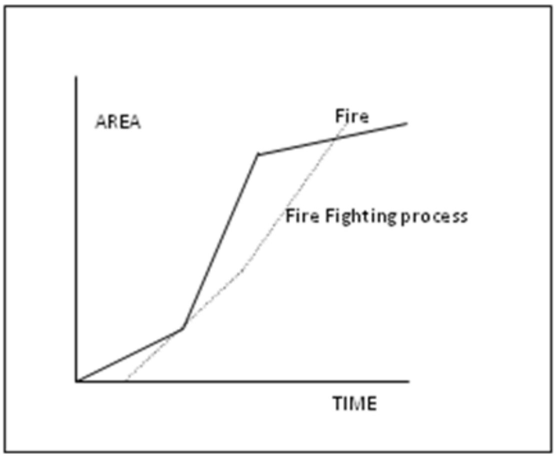



**Figure 3.1: Example of Variable Training Program Trial**

### 3.1.4 The form of the dataset

Data from the Cañas et al. (2005) study comprises precise protocol information about the nature of every command issued by the participants for each trial and details of the trial's configuration; this information is stored in a set of 'history' files. Figure 3.2 shows the first ten commands issued by a participant; this is an excerpt from a FireChief history file (subsection B in section 2.3.2.5 describes each column in the figure). In this example the first command was issued at generation 19, no terrain cells have been destroyed so performance is still 100%, it was a *Move* command that resulted on the displacement of Copter (number 4) from (11,4) to (13,10) which is of terrain type Pasture.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | Move | 19 | 100.00 | 4 | Copter | (11, 4) | (13, 10) | Pasture |
| 2 | Drop Water | 31 | 100.00 | 4 | Copter | (13, 10) | | Pasture |
| 3 | Move | 54 | 100.00 | 3 | Copter | (8, 6) | (11, 11) | Clearing |
| 4 | Drop Water | 63 | 99.77 | 3 | Copter | (11, 11) | | Clearing |
| 5 | Move | 82 | 99.42 | 2 | Truck | (4, 11) | (13, 12) | Clearing |
| 6 | Move | 109 | 98.95 | 4 | Copter | (13, 10) | (12, 10) | Clearing |
| 7 | Drop Water | 111 | 98.95 | 4 | Copter | (12, 10) | | Clearing |
| 8 | Control Fire | 118 | 98.71 | 2 | Truck | (13, 12) | | Clearing |
| 9 | Move | 133 | 98.71 | 1 | Truck | (4, 14) | (13, 13) | Forest |
| 10 | Move | 148 | 98.71 | 4 | Copter | (12, 10) | (12, 9) | Forest |

**Figure 3.2: An excerpt of a FireChief history file**

## 3.2 Analysing Strategic Behaviour

Siegler & Shipley (1995) and Lemaire & Siegler (1995) identified three dimensions related to strategy use. The first one is strategy variability in the sense that that there exist multiple strategies for solving, for instance, the FireChief task (refer to section 3.3.2). The second dimension is the strategy base rate of success which is related to the relative frequency of use of each strategy and the types of problems for which they are used (cf. section 3.4.1). The third dimension of strategy use is strategy change: improvements in the way a strategy is implemented can occur due to practice. Siegler & Shipley (1995) developed a model of strategy choice called ASCM (pronounced "Ask-em"): Adaptive Strategy Choice Model where information about the solution, the time spent in solving the problem and the accuracy of the strategy used modify the perception about the strategy and the problem. ASCM was applied in the context of addition problems and two strategies were identified: direct retrieval of the answer and the computation of it. In ASCM strategy selection is based on the projected strength of each strategy in relation to the current problem. The authors argue that "problem-specific information based on a substantial database is the best predictor of a strategy's future effectiveness on that problem." (p. 28). The ASCM model gives more weight to recent performance. When a problem is presented a stepwise regression equation computes the strength of each strategy for the problem (these ideas are similar to the concept of utility used in ACT-R). According to ASCM strategy execution improves with practice because there is an increase in its execution speed and a decrease in its probability of generating an error. With enough practice a strategy may be consolidated: its implementation becomes smooth and rather unreflective.

Siegler & Shipley also noticed that it is "surprisingly difficult to override usual selection through metacognitive means…" (p. 14). They also stress the difficulty associated with learning not to use old strategies. These findings are related to the cognitive inflexibility reported by Cañas et al. (2005). The ASCM model is able to improve performance because, as more arithmetic problems are completed, the answers to these problems (obtained either by retrieving the answer or calculating it) increase their associative value with the problem (generating a peaked distribution), so eventually the direct retrieval of the answer happens more often. In FireChief performance improvement is significant too (see section 3.4.4). Consequently, it is hypothesized that participants in the VT group are able to choose better strategies during the test phase because they are more flexible. It is also hypothesized that people in the CT group execute strategies more quickly and more effectively as more trials are completed due to strategy consolidation, but that their knowledge will be limited to fewer strategies.

### 3.2.1 The Cañas et al. (2005) strategies

The definition of a coherent set of strategies represented a challenge mainly because the FireChief environment changes independently of the actions of the individual and the available actions at each simulation step are numerous. The first step taken to arrive at a set of strategies was to look at the results reported in the Cañas et al. study. The two strategies described originally in the work of Cañas et al. (2003) and used again later in the Cañas et al. (2005) work are the Move-Drop and Control strategies. In the Move-Drop strategy participants move appliances to the closest unattended fires and drop water there. Trucks are not sent to fires that are too fierce and where they could be destroyed. The Control strategy involves finding the closest fire and then sending an appliance to execute a CF command two cells away from it in a randomly chosen direction. In the Cañas et al. (2005) study there is a further division of the Move-Drop strategy into the Frequent-Move-Drop and Infrequent-Move-Drop strategies. These strategies differ in the number of DW commands emitted but the pattern of commands (represented by a transition matrix) is similar. Although the strategies found by these authors served as a starting point a richer set was found through a new analysis described later. In a previous study (Cañas et al., 2003) the authors present an algorithmic description of a couple of strategies, these strategies are described below.

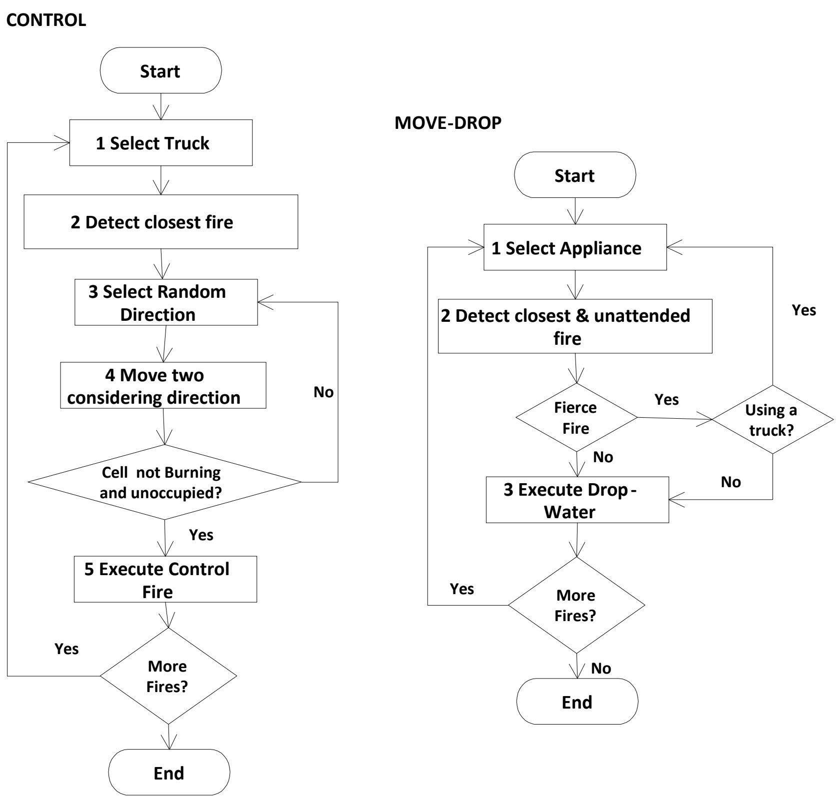



**Figure 3.3: FireChief strategies inspired in Cañas et al. (2003)**

Two flow diagrams depicting the strategies from Cañas et al. (2003) are presented in figure 3.3. These flow diagrams were constructed following the textual description of the strategy; they are presented with the aim of showing that the Control strategy requires extra decision making steps than the Move-Drop strategy and for making a comparison with the new set of strategies found in this study. It is possible to see in figure 3.3 that in both strategies a unit must first be selected (in the case of the Control strategy it must be a truck). In the next step of both strategies a fire must be selected. The strategies then differ from the third step onwards. In the Move-drop strategy there is perceptual action that could produce two outcomes depending on fire intensity: either a drop water command is executed if the fire is not too strong or a new unit is selected if the selected unit is a truck. In the Control strategy there are more steps: a random direction for executing the Control Fire command is selected and the resulting location is shifted two cells to give a distance between the fire and the targeted cell. After that a perceptual action is executed over the target cell to determine if it is suitable. If the target cell is adequate a Control Fire command is executed; if not a new random direction is selected.

#### 3.2.1.1 Transition matrices

The use of transition matrices for studying problem solving behaviour was originally proposed by Howie and Vicente (1998) and was adapted by the Cañas et al. (2005) study. A matrix of transitions is obtained for each FireChief protocol. Rows and columns of the matrix represent the full range of commands, and the cells contain the number of times one type of command follows another (see table 3.6). According to Howie and Vicente (1998) transitions between actions reflect the order in which a person issues the sequence of commands and this is more informative that just their frequency. This approach proved to be useful to detect differences in strategy use (see section 3.4.2). Nevertheless, a description of behaviour provided by a transition matrix is very abstract when the goal is to develop a cognitive model where behaviour is reproduced at the level of milliseconds. Given that FireChief commands take seconds to be executed, there are a considerable number of cognitive, perceptual and motor actions taking place in the execution of each command; and this information is not provided in a transition matrix. In order to classify participants according to strategy use, Cañas et al. started by identifying pure strategies (described in figure 3.3) and determining the degree to which participants present each of these pure strategies using transition matrices analysis (Antolí, personal communication, May 2009).

### 3.2.2 The Protocol Analysis Tool (PAT)

To assist with the analysis of protocols, a software tool called the Protocol Analysis Tool (PAT) was developed in Visual Basic.NET. The main input of the system is the history files generated by the FireChief simulation (figure 3.2). PAT populates and uses a comprehensive data model (figure 3.4) able to represent detailed FireChief performance. PAT enables the processing of all the trials for both participants and the cognitive model. Having in mind the goal of identifying strategies, PAT was extended to support the generation of plots of the problem solvers' time-contingent commands, the generation of transition matrices (section 3.4.2) and metrics extraction (section 5.2).

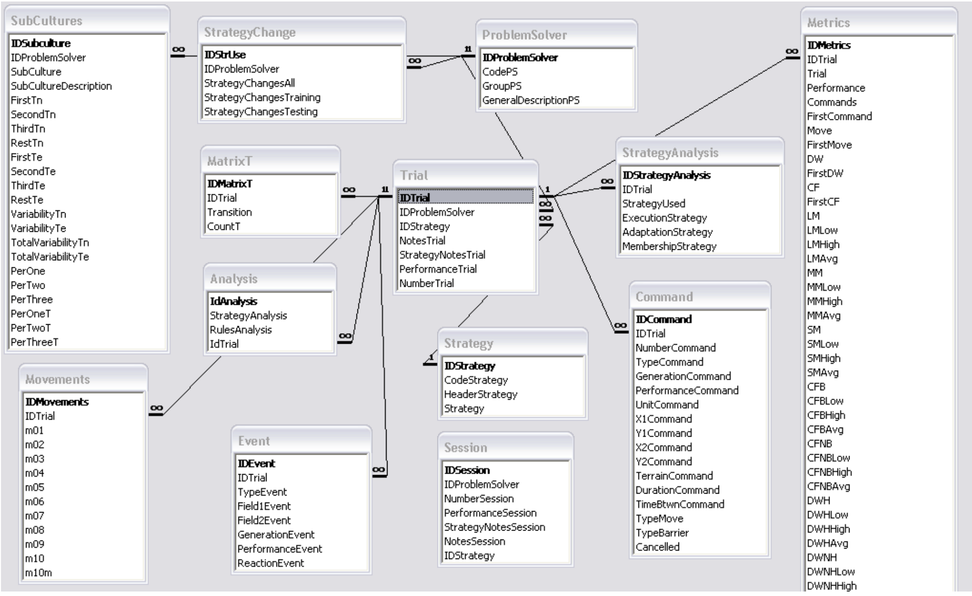
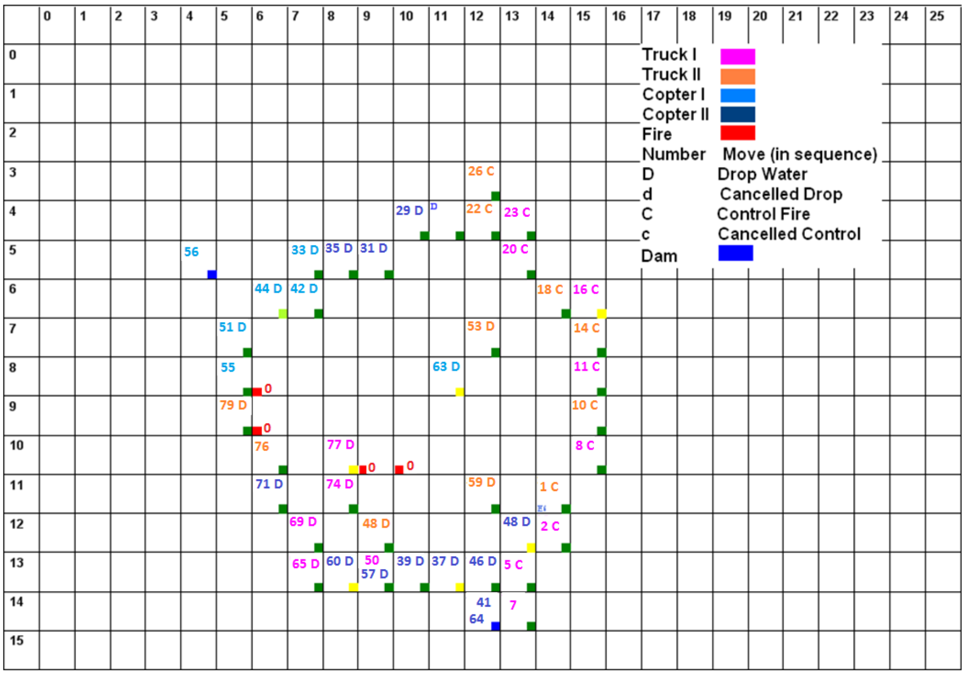



**Figure 3.4: PAT's database (main tables)**

One type of view depicts all the commands with their position within the landscape with their sequence number, type and unit. If many commands were issued in the same cell the sequence is preserved. The type of landscape is represented using different colours and environmental changes, such as wind direction and strength, are also shown.




**Landscape elements (coloured squares)**
**Yellow: clearing**
**Green: forest**
**Light green: pasture**
**Pink: house**
**Blue: dam**
**Red: Fire**

**Units (coloured writing):**
**Pink: Truck 1**
**Orange: Truck 2**
**Blue: Copter 1**
**Dark Blue: Copter 2**

**Figure 3.5: Example of a view. The coloured writing shows two pieces of data: the generation at which the command was issued and the type of command (D for Drop Water and C for Control Fire)**

Figure 3.5 shows an example of this view. The position of the initial fires is shown using red squares; each unit's behaviour is distinguishable by a colour and different letters or numbers distributed throughout the landscape (in figure 3.5 it is possible to observe all the movements made by truck one by following the sequence of numbers in pink and the different commands issued by this unit). Within this view it is also possible to distinguish cancelled commands (a lowercase c/d represents a cancelled Control Fire/Drop Water while a capital C/D represents a successful Control Fire/Drop Water). All the numbers in this view are timestamps expressed in generations. The coloured squared at the bottom right of the cell represents the terrain. There is another kind of view that shows the distribution of commands in the landscape, their type and unit but also shows a timestamp for each command, the initial fire position and environmental changes (see figure 4.11).

### 3.2.3 Methodology

Using the PAT it is possible to gain an understanding of the patterns of commands issued by the participants providing some insights into the strategies implemented by them. In the initial phase of analysis, emphasis was put on creating a repertoire of strategies with the aim of later classifying all the protocols under this structure. Strategy definitions were constructed following two steps: (1) grouping similar patterns observed in the data and (2) constructing a specification at a level of command executions that could be translated into ACT-R commands. In order to classify a trial's protocol, the collection of commands executed by a participant during a FireChief trial is loaded into PAT. The analysis conducted for every protocol produces a description of the pattern of commands observed during that trial. Various aspects of the data were considered when conducting this analysis. Typically, participants seem to have a priority guiding their behaviour which is catalogued for example as "create a barrier" or "attack fire". They also have an initial focus of attention such as "the strongest fire". If CF commands are used, their distribution and pattern of use is catalogued, for example: "a line of CF commands is created from north to west", "CF commands are interleaved with DW commands…" and so on. These observations can be confirmed with other metrics, a transition matrix can also show if a participant is interleaving CF with DW commands. Stages of the problem solving activity are also identified, for example: "when a fire is under control". Observer remarks about unit use are recorded: "only copters are used to issue DW commands". Changes of strategy are also identified: "frequently the fire is not stopped by an incomplete line of CF commands; in this case a common adaptation is to follow a similar pattern that resembles following the fire's path". Observations about strategy implementation are made: "it is hard to implement this strategy when appliance efficiency is low". To assist the process of analysing a protocol its corresponding replay files (also provided with the original data) can be used; this was very useful when the visualizations were not sufficient for understanding participant intentions. During this process it was also necessary to comprehend the different FireChief trials (table 3.2) to allow a better understanding of events and processes. The intention of the analysis was to arrive at qualitative descriptions of patterns of behaviour used by participants with the ultimate goal of using them in the implementation of the cognitive model. At a later stage these patterns of command use were compared by using the type and the spatial/temporal distribution of commands for creating the new hierarchy of strategies presented here.

The first group of participants that were analysed was constituted by the five participants of group CTW who showed the highest overall performance. The rationale behind this decision was that it was easier to find discernible patterns among trials that ended with good performance (i.e. the problem solver´s deployment of resources in order to fight the fire was clearer). Another reason is that there is every reason to believe that the high performers were more motivated to 'solve the problem' and therefore the strategic problem solving behaviour of interest to this research is more likely to be evidenced in these protocols. The next step was to analyse the five highest performers for the remaining groups.

At the end of this process 480 protocols were analysed and the result was a set of 26 distinguishable patterns (i.e. strategies). The next step was either to assign the remaining 1280 protocols to each of the established categories or, if a new strategy appeared, add it to the repertoire. It is important to point out that this classification was carried out by a single observer, a cross validation of the results of the classification would improve the quality of this process. Nevertheless the identification and labelling of specific strategies does not have significant consequences for the key research questions related to strategy use, consolidation and adaptivity, particularly about how choices arise in complex dynamic tasks. The interesting part of this research is how these features are produced and how strategy execution is affected by environmental feedback. Chapter 6 demonstrates that the patterns found during this analysis are reproduced by a model that follows the strategy definitions described in this chapter, in other words, that by having the intentions defined by a strategy the resulting pattern is what is observed in the empirical data, including the considerable amount of variability.

## 3.3 Hierarchy of strategies

The goal in FireChief is not clearly defined so it needs to be decomposed and further specified. Participants are instructed to stop the development of the fire and hence the problem solver is not biased towards fighting the fire directly or issuing *Control Fire* commands to halt its future development. In general terms, the way of deploying resources in the landscape defines a strategy. The hierarchy of strategies depicted in figure 3.6 is a product of the brand new analysis conducted in this research.

In order to minimize subjectivity during the process of finding strategies the following mechanisms were implemented. First, the strategy patterns found in the Cañas et al. (2005) study were considered. As two distinguishable clusters emerged from the reanalysed data, one group in which problem solvers use a mixture of DW and CF commands and a second one in which problem solvers use DW commands primarily, it was decided to re-use this classification as the second layer in the hierarchy (the names were altered: Move-Drop to only-DW, and Control to CF-DW). Second, for the third level in the hierarchy four criteria were applied: command frequency, command sequencing, spatial distribution of commands and temporal distribution of commands. For command sequencing transition matrices were created for each proposed strategy (see table 3.4) the other 3 measures were extracted by querying PAT's database. Groups with similar patterns considering these criteria were obtained from the database. Significant spatial (less than 2 cells) and temporal proximity (less than 2 seconds) between commands were used to identify those strategies that belong to the Barrier and Stop strategies. Using this information the third layer of figure 3.6 was created consisting of the Barrier, NonBarrier, Stop and Follow strategies. Third, to define the fourth layer in the hierarchy, every protocol was visualized using PAT to discover differences based on specific spatial distributions of commands that were not detected by the current functionality of PAT such as distinguishing between a circle, a line or a semicircle pattern of CF commands. During this stage those strategies that make use of CF commands were primarily characterized by the way in which these commands were used. Fourth, the criterion to limit the number of leaves in the hierarchy tree was the existence of at least five cases of a particular pattern of command use executed by at least two participants. In total 26 different strategies were found. After the hierarchy was completed, all participants' trials were classified as members of one of the strategies at the bottom layer of the hierarchy.

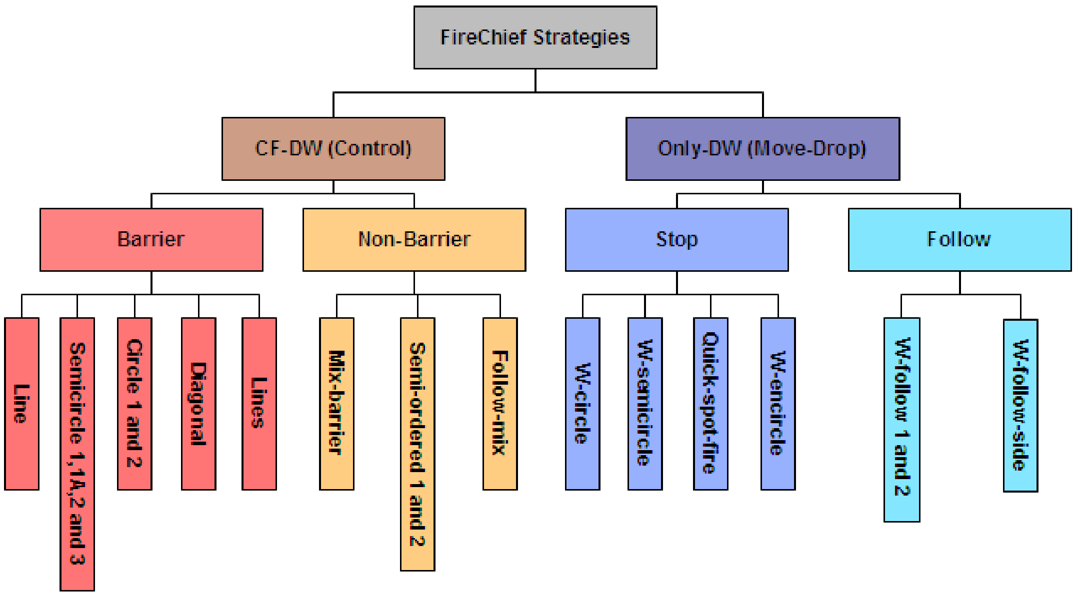



**Figure 3.6: Hierarchy of strategies**

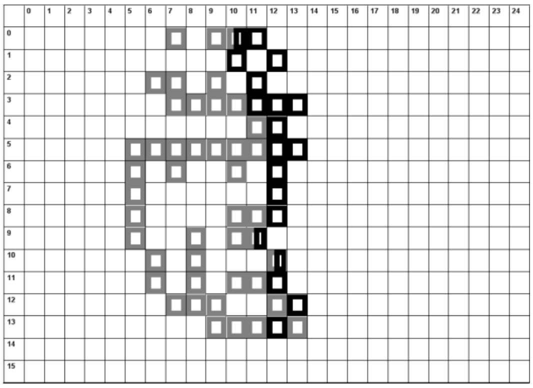
### 3.3.1 The Barrier strategy

The Barrier strategy presents a very characteristic way of dealing with the fire: the issuing of an ordered pattern of Control Fire (henceforth CF) commands (the appearance of this pattern is similar to a barrier) which may stop the fire's development. If the wind changes its direction, this barrier will not be effective. The existence and the form of this barrier is the most salient aspect of this strategy. There are many forms in which the barrier is created but a semicircle and straight line are the most frequent. The selection of a barrier form represents behavioural differences that will ultimately yield differences in the way fire is controlled (or not). The effectiveness of a barrier form depends on many factors: the morphology of the fire, the wind conditions and the implementation of the barrier. Graphical examples of the described strategies are provided. In these graphs a grey rectangle represents a cell in which a Drop Water (henceforth DW) command was issued and a black rectangle represents a cell in which a CF command was executed.

### 3.3.1.1 Barrier subtypes

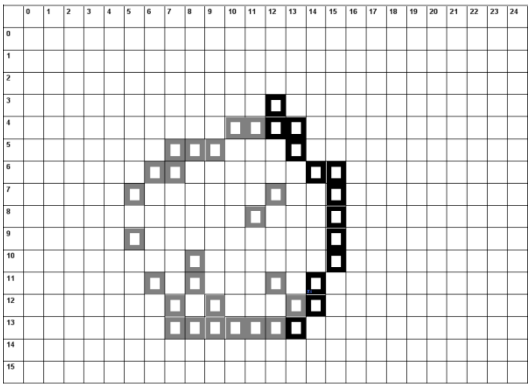
The Barrier strategy has nine main subtypes which differ among them in the shape of the barrier and in the ratio between CF and DW command use.

## A) Line - a line of CF commands forming a barrier.



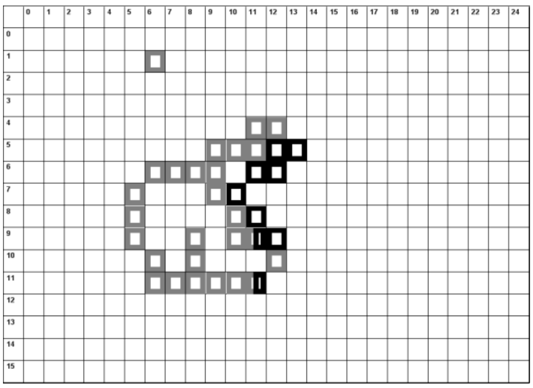
The barrier is created in the direction of the wind in a pattern similar to a line. It is good practice to use both trucks during the creation of the barrier. When the fire is under control the remaining fire is attacked. In this strategy the CF commands have the distinctive function of stopping the fire and the DW commands have two functions: to gain enough time to construct the barrier and to combat the remaining fire. This strategy can yield high performance.

## B) Semicircle I - a semicircular barrier of CF commands complemented by a semicircular line of DW commands.



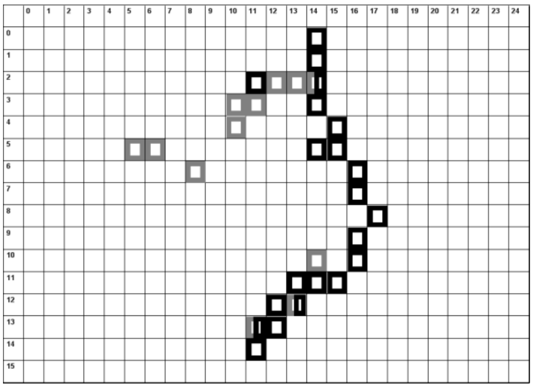
The idea is to stop the development of the fire with a barrier of CF commands and then to attack the remaining fire by dropping water. There is a clear priority of the creation of a semicircular barrier (frequently this barrier starts at the south-east of the fire when the fire blows to the east). After the barrier is finished the fire is fought using the copters in a semicircular fashion starting from the north and finishing at the south. The fire in the middle of the circle spreads freely. The result is a circle that has a high concentration of CF commands in the East and a high concentration of DW commands in the west. The diameter of the circle varies according to the wind strength (e.g. a wind strength of 4 generates a small circle and a wind strength of 6 creates a large circle).

## C) Semicircle 1A - a segment of a semicircular barrier is created with considerable use of DW commands.



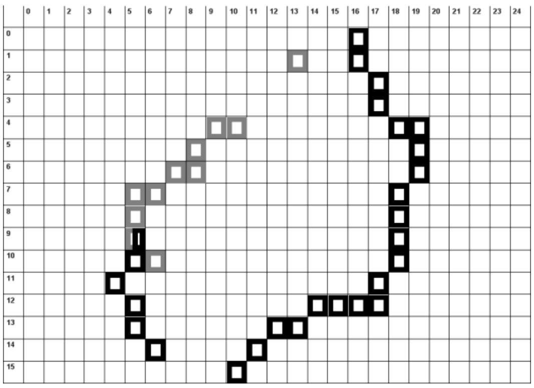
A greater proportion of DW commands are used in this version of the Barrier strategy compared to those above. Using as a reference the constant training scenario, fire is first attacked to the east of the eastern fire creating a semicircle. After this first attack, the fire is attacked toward the west (if the strategy is successfully applied the fire is stopped and cannot spread further eastward) so that all units drop water creating a circle around the original fires. A distinguishable barrier of CFs is created in one of the quadrants of the protective circle in an arched fashion. The order of cells targeted in the formation of the circle usually takes into account changes in the behaviour of the fire (wind direction) so it is not always implemented clockwise. An advantage of this strategy over W-Circle (see section 3.3.3.1) is that the barrier of CFs is often created at the same time as the circle of DWs employing more units at one time and thus exploiting the available time.

### D) Semicircle 2 - a semicircular barrier of control fires with low use of DW commands.



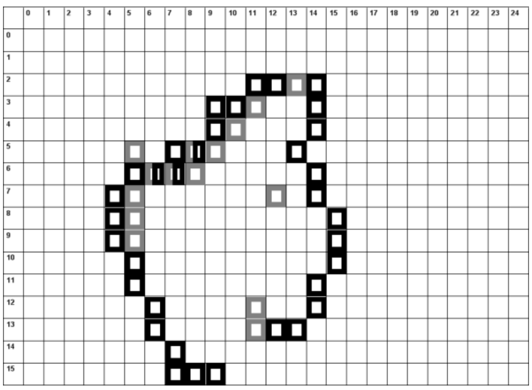
A distinguishable barrier of CFs is created in one of the quadrants of the protective circle in an arched fashion. It is better if both trucks are used in this process (the distance to the original fire can be reduced) but there may be a preference for the use of one unit. The barrier creation process takes a considerable time (usually 800 generations). Most of the DW commands are issued close to the already partially formed barrier. After the barrier is finished a modest amount of DW commands are issued to combat the remaining fire.

### E) Semicircle 3 - segments of curved barriers supported by DW commands.



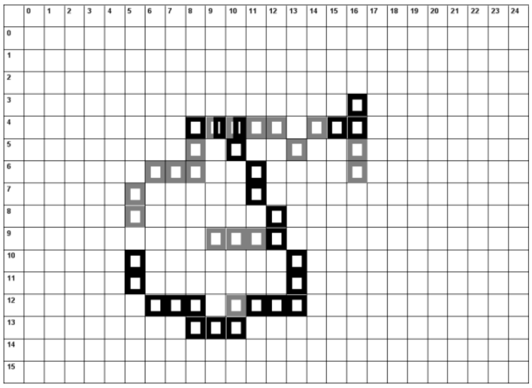
There is a clear priority on the creation of a curved barrier. Usually both trucks are used in this process. The curved barrier is not as long as the one created when implementing Semicircle 2. Other curved barriers are created at the circumference of the fire (but the final shape is not always circular). The barrier creation process lasts almost all the trial and the use of DW commands is very low.

### F) Circle 1 - a circular barrier of CFs around the fire.



A full circle of CFs is created around the original fire taking into account the initial wind direction. A few DW commands are issued serving to give time to construct the barrier. Usually fire spreads freely in the centre of the circle but sometimes (if there are no spot fires) the inside of the circle can be protected. This strategy can be very effective when the wind changes direction.

### G) Circle 2 - a nearly complete circular barrier of CFs around the fire plus a segment of DW commands.

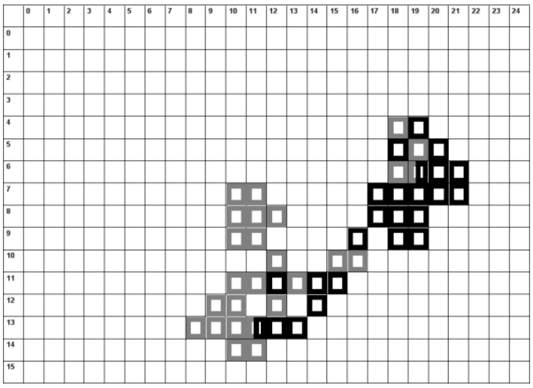


As in the Circle 1 strategy almost a full circle of control fires is placed around the original fires; the difference is that in some regions a curved section of DW commands are issued and so the CF barrier skips these regions. Usually the fire spreads freely within the circle.

#### 3.3.1.2 Other frequent patterns

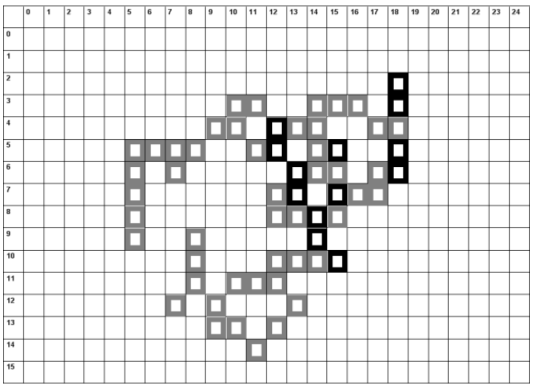
In the following cases various attempts to stop the fire are made in the form of many barriers. Presumably the problem solver is trying to stop the fire with a single barrier but the fire is able to bypass it so new barriers are created in different locations.

### A) Diagonal 1 - a diagonal barrier of CF supported by DW commands.



Usually fire is attacked near its origin. A barrier of CFs is created following a diagonal pattern from north-west to south-east. Both copters fight the fire under this barrier. This barrier is very effective in controlling the first change of wind condition (from east to north-east) but neglects the remaining wind changes. Other diagonal barriers can be created to control other developments. The DW commands can also serve to give extra time for the creation of the barrier.

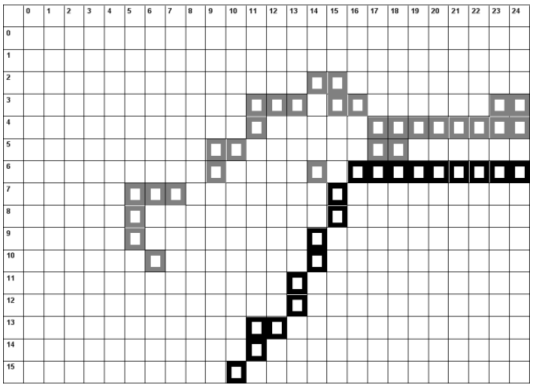
### B) Lines - a set of linear barriers created one behind the other supported by DW commands.



The focus of the strategy is in creating lines of CFs in the direction of the wind. The lines can be created from north to south or vice versa (considering that the wind blows east). A few DW commands are issued. If the fire bypasses the first line a second one is created. Usually only copters drop water. Lines can be straight or curved. This strategy is usually deployed with poor results (there is no real control of the fire).

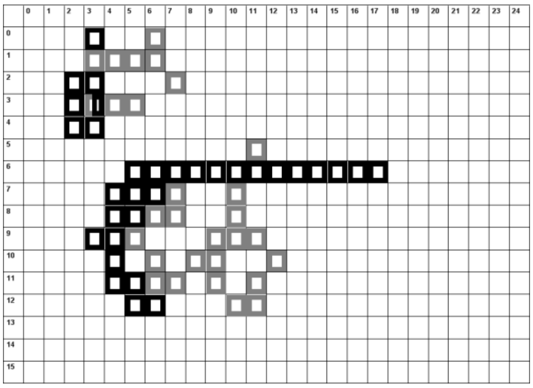
The majority of trials catalogued under Barrier fall among one of the seven subtypes described above, but there are some trials where, although a barrier was created, it was not possible to categorize it as a subtype nor did it appear with enough frequency to justify a new subtype.

### C) Semicircle and horizontal line - a curved barrier of CF followed by a horizontal line supported by DW commands.



A semicircular barrier is created from the SE towards N (when the wind direction is East). After reaching the north of the original fire the barrier changes direction creating a horizontal barrier towards the east. Fire is fought in a westerly direction (often with a similar DW pattern to that of Semicircle 1) above the horizontal barrier using all units. The region at the east of the initial fire can also be protected with DW commands. Another pattern of Drop Water commands is to create a semicircle from the South towards the West (ignoring the Northern section of the second part of the barrier). A hypothesis is that the second part of the barrier is created to protect a region with houses. Only participant A19 executed this version of Barrier.

### D) Line West-East - a line of CFs in the direction of the wind plus a curved line of CFs in the opposite direction of the wind supported by DW.



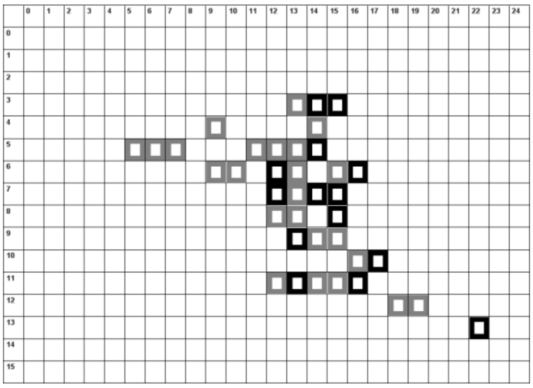
There is a focus on the creation of a long horizontal barrier above the initial fire by alternating the use of the two trucks. The initial fire spreads freely towards the east, during the first generations of the trial, but the subsequent shifts of wind direction guides the fire front towards the barrier. Another section of the barrier is created at the western end of the initial barrier (in a semicircular fashion) in order to anticipate the last shift in wind direction. The rest of the fire is attacked to the east of this barrier typically using only copters. This strategy can be very effective when the wind direction changes and was observed in one participant only: A02.

### 3.3.2 The NonBarrier strategy

The noticeable dispersion of CF commands is the most salient characteristic of this strategy. Presumably there is less use of cognitive and perceptual actions for selecting a cell for executing the CF command so it takes less time to place it (the cognitive model explores this issue); the drawback is that the CF commands are seldom placed in a way that they can effectively control the development of the fire. In this strategy CF commands are alternated with DW commands. The CFs are placed not randomly but in consideration of the development of fire, but allowing a little extra distance between the location of the CF command and the fire in comparison with the Barrier strategy. However, the choice of location for the CFs is not always the most useful. In general terms versions of the NonBarrier strategy use only 66% of the amount of CF commands used by versions of the Barrier strategy.

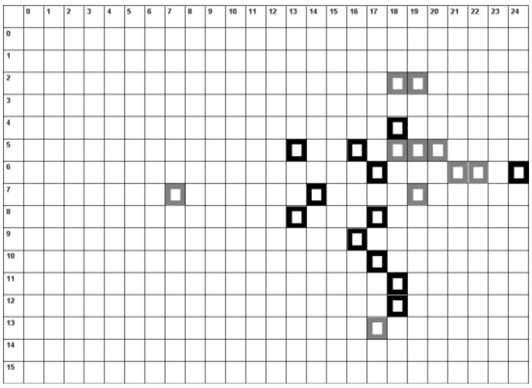
### 3.3.2.1 NonBarrier Subtypes

## A) Mix barrier - a mixture of CF and DW gives the impression of a barrier.



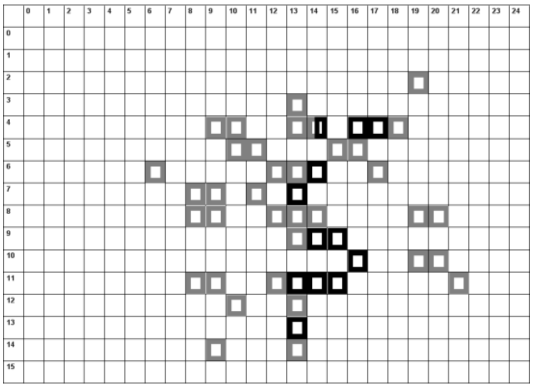
There can be a priority on attacking the fire in positions near the initial eastern fire before creating the barrier. CFs are deployed in the direction of the wind in a certain order (usually from south to north, following a line pattern) CF commands are interspersed with DW commands in order to create a barrier. After the fire is under control the remaining fire is usually attacked with DW commands. Frequently only copters are used to issue DW commands (it is better if copters are used alternately and to appropriately refill the tanks). Frequently the fire is not stopped by the incomplete line of CFs, in this case a common adaptation is to follow a pattern similar to W-follow 1 or W-follow 2 (see below) to combat the fire. It is hard to implement this strategy when the appliance efficiency is low: the fire will not be extinguished by the DW commands and will spread across the barrier.

## B) Semi ordered 1 - a semi ordered pattern of CF supported by DW commands.



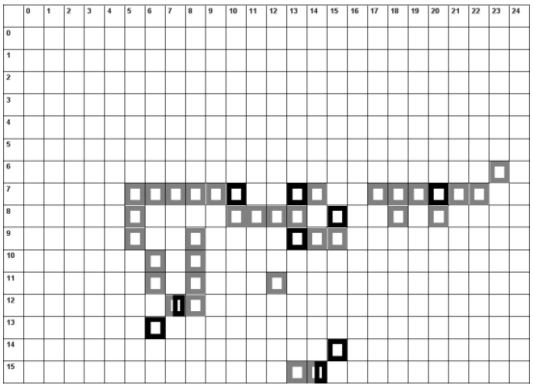
Cells on fire close to the initial fire are usually attacked with DW commands during the first few seconds of the trial. CFs are deployed in the direction of the wind in a certain order (leaving spaces, usually from south to north, following a line pattern). Although the fire bypasses the barrier there is more interest in fighting the fire at the left side of this barrier (where fire strength is low). Frequently the number of Drop Water commands is low. This strategy also includes trials where incomplete barriers of CFs are created with no support by DW commands. The usual pattern of the barrier is a line but also semicircles can be created. Performance is never high due to the spreading fire.

### C) Semi ordered 2 - a semi ordered pattern of CF supported by DW commands.



Cells on fire close to the initial fire are usually attacked with DW commands at first. CFs are deployed in the direction of the wind in a certain order (leaving spaces, usually from south to north, following a line pattern). Although the fire bypasses the barrier there is more interest in fighting the fire at the left side of this barrier (where fire strength is low) frequently the number of DW commands is low. The main difference with Semi-ordered 1 is the thickness of the barrier; in this case the barrier is created by (horizontal) lines of two or three CFs. It is also harder to identify a linear or semicircular pattern. This strategy appears less frequently than Semi-ordered 1.

### D) Follow mix - follows the development of fire with a mixture of DW and CF commands.



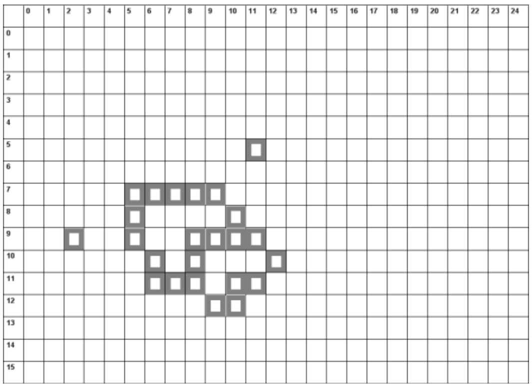
A mixture of commands is issued but there is always more DW than CF commands. The fire could be closely attacked if the wind strength is low. After some time all units are fighting the fire. When the fire spreads from the centre different units fight different regions of the fire. There is considerable command dispersion; the fire is not attacked systematically so it covers extensive terrain. This strategy is a reactive one in the sense that it follows the development of the fire rather than attempting to control it. Commands may be issued in a pattern of lines.

## 3.3.3 The Stop strategy

The systematic use of DW commands which may stop the fire is the characteristic feature of this strategy. In this strategy DW commands are issued on the most intense fires and are issued in sufficiently close proximity to each other to stop the development of regions of fire. High wind strength makes the implementation of this strategy difficult. In the constant training group this strategy is not often used. By observing the protocols, and later by running the model (section 5.2.1), it was possible to see that the success of the Stop strategy depends upon the precise sequencing of actions. For example, there are two strategically-important fires (these fires are the first to destroy a cell and expand) at the beginning of the constant training trial; if these two fires are attacked quickly enough the trial can be solved more easily. Another important feature of the landscape is that the most intense fire will inevitably encounter a clearing in its path reducing its intensity, so the best approach is not to waste time attacking it but instead to wait until it arrives at the clearing.

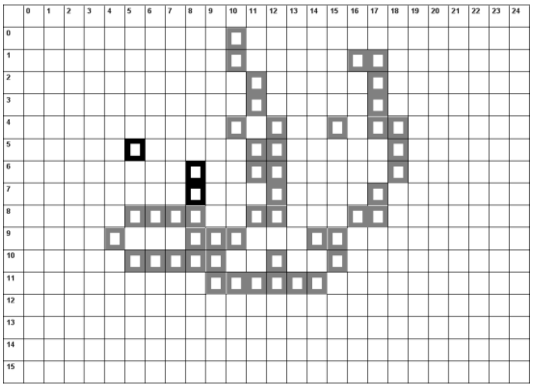
### 3.3.3.1 Stop Subtypes

## A) W-circle - creates a circle of DWs around the fire.



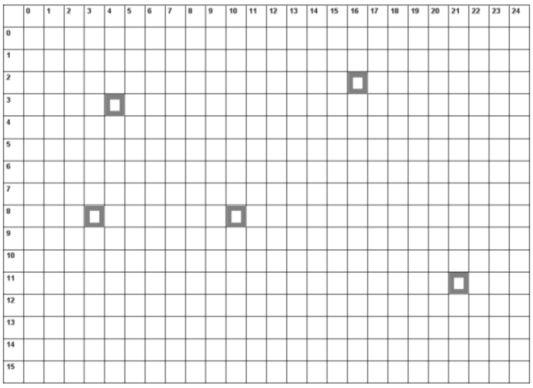
This strategy uses mostly DW commands. Using as a reference the CT scenario, fire is first attacked at the east of the eastern fire creating a semicircle. After this first attack, the fire is attacked towards the west (if the strategy is successfully applied the fire is stopped and cannot spread further eastward) so that all units drop water creating a circle around the original fires. This strategy can be highly effective and stop almost any development of fire. If this strategy is successfully implemented new spot fires will appear early in the scenario imposing new demands on the problem solver. The dynamic of fire development generates a rounded shape of fires so the Stop strategy must issue DW commands over this circle of fires. If the wind strength is 3 or 4 the diameter of this circle is of 3 or 4 cells wide; a wind strength of 5 or 6 produces a circle of 7 or 8 cells, making it very hard to implement this strategy. When there is only one mature fire to attack, the application of this strategy is easier and on many trials in the variable training condition problem solvers are able to apply this strategy successfully many times. This strategy yields very high performance.

## B) W-semicircle - create one or many curves of DW to stop fire development.



This strategy uses mostly DW commands. The characteristic feature of this strategy is that one or more arched DW barriers are created around the fire. Units are deployed to different surrounding regions using DW commands to try to stop the fire. Frequently the first arched barriers are very close to the original fires. This strategy also appears when many DW commands are issued to create a thick semi-circular barrier of DW commands. Often, when there are changes in wind direction, arched barriers of DW commands are created starting from the centre of the fire. The density of DW commands around the original fire varies. Usually many curves are created around the original fires, if the fire escapes then it is followed and again captured within them. This strategy focuses on stopping fire development (this distinguishes it from W-Follow 1, section 3.3.4.1-A). Another distinction is that the semicircles are created one after another (a distinction with W-Encircle, section 3.3.4.1-D, where there is no clear order in the creation of the barriers, and with W-Follow 1 where the fire is followed in different regions at the same time).

## C) Quick-spot-fire - stop a fire in its starting cell by using a DW.



This strategy uses only DW commands. It can only be used for spot fires which the problem solver needs to attend to with enough speed so that its development is halted in the original cell. Wind strength is critical for performing this strategy: if wind strength is higher than 5 the spot fire will develop before the problem solver can attend to it (this can be seen by comparing trials 5 and 15 of the variable training set). If this strategy is applied successfully it can yield perfect scores on many variable training trials.

## D) W-encircle - issue DW commands around the fire without apparent order.

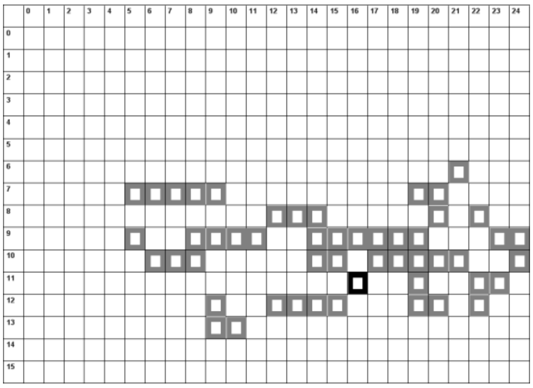
This strategy is similar to a W-Follow 1 strategy, but the problem solver uses a distinguishable pattern of lines to attack the fire. These lines are placed around the fire, so the central part of the landscape is not protected. The main difference with W-Circle 1 is that there is no systematic use of commands; rather they are placed at different points of the final shape so there is activity in different areas of the landscape at any given time. This strategy combines following the development of the fire and trying to stop it.

### 3.3.4 The Follow strategy

The strongest fires are not considered at the beginning of the trial so the focus of attention is on the weakest fires. A characteristic of Follow is that there is a moment during the running of the trial in which the fire is allowed to burn and to develop to a certain extent. In this way the behaviour adopted by the participant is to follow the fire's development. Copters and trucks are used when implementing this strategy but using trucks represents a higher risk because the fire is developing continuously and can engulf the trucks. The strongest fires may be considered but the whole behaviour looks as if the participant is following the development of the fire. It is frequently the case that in its latter stages the Follow strategy follows a process in which advanced fires (considering the direction of the wind) are sought and DW commands are used to extinguish them. An advanced fire usually has a high strength. This strategy is less structured because the selection of fires to attack is not as systematic as in the case of the Stop strategy.

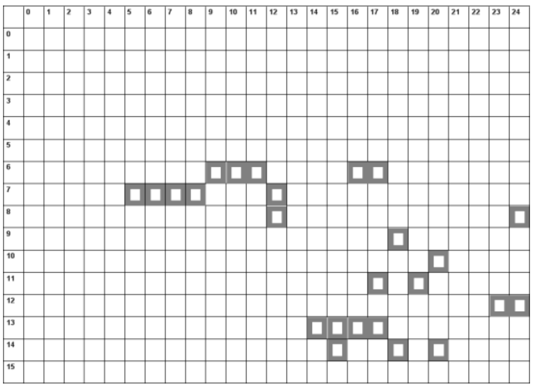
### 3.3.4.1 Follow Subtypes

## A) W-follow 1 - follows the development of fire with frequent use of DW commands.



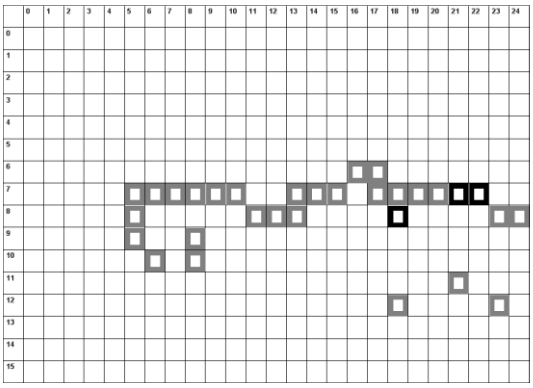
First the closest positions to the eastern fire (or the western, but not both) are attacked using copters. After some time all units are fighting the fire. When the fire spreads from the centre different units fight different regions of the fire. Copters are used more than trucks and are sent to the furthest regions. There is a clear order when attacking the fire with the sense that some units work together to extinguish the fire in certain zones. This strategy is highly adaptive to changes in the wind: the units are sent to where the fire is. The fire is (usually) not stopped but much of the terrain that catches fire is saved from destruction. The fire continues its path but the amount of lost terrain is not much. Usually there is a space in the middle of the lines of DWs that is not attacked, and that allows the development of the fire. At other times the pattern of DWs is like a uniform block following the direction of the wind, so the normal pattern for individual units is a line following the wind direction. This strategy may be used when the fire cannot be contained in the centre. It seems to be more effective if only copters are used.

## B) W-follow 2 - follows the development of fire with less frequent use of DW commands.



Only (or mostly) DW commands are used. First the closest positions to the eastern fire (or the western, but not both) are attacked using copters. After some time all units are fighting the fire. When the fire spreads from the centre different units fight different regions of the fire. Copters are used more than trucks and are sent to the farthest regions. The difference with W-Follow 1 is that there is more command dispersion: there is a greater distance between the clusters of DW commands and the frequency of them is less. The fire is attacked less systematically and hence it covers a lot of terrain. This strategy is a reactive one and performance is usually poor.

**C) W-follow-side - follows a distinguishable section of the development of fire with frequent use of DW commands.**



Mostly DW commands are used in this strategy. In the first stage of the strategy DW commands are issued in different regions of the fire. When the fire starts developing, one side of the spreading fire is chosen (north or south if the wind is blowing east) and the DW commands are focused on that part. The difference with W-Semicircle 1 is that in this case some parts of the expanding fire are ignored and there is only one visible line. In this strategy there is less dispersion of commands due to this focus on one side. The DW commands are issued at the border of the moving fire. Another difference with W-Semicircle 1 is that the DW commands follow the development of the fire instead of trying to stop it.

### 3.3.5 A comparison of strategies

Patterns of command use can be compared by using the type and spatial/temporal distribution of commands. The first basis for comparison is the use of the CF command, specifically the degree of structure in the use of this command. Degree of structure is a function of the spatial and temporal proximity of commands and the existence of specific patterns such as lines and semi-circles. If we were to plot all strategies on a continuum representing the 'degree of structure' dimension then, in the centre of this axis we would place the Semi-circle 2 strategy and in the extremes the highly-structured Circle 1 and relatively unstructured Follow-mix 1 strategies. While a full circle of CF commands is created around the fire in Circle 1, in Semi-Ordered 1 CFs are issued leaving a noticeable distance between them.

Another criterion refers to the degree of structure in the issuing of DW commands, particularly their spatial proximity. The W-follow 1 strategy tracks the path of fire development and extinguishes as much fire as possible by a systematic use of the available units while W-follow-side focuses on a particular sector of it. Differences in the use of the DW command generate different versions of the Barrier and NonBarrier strategies. For example, the main difference between Semicircle 1 and Semicircle 2 is that in the latter there is no use of DW commands to extinguish the fire. Two criteria govern the differences among Only-DW strategies. The first is whether the strategy appears to try to stop fire development or not. In the case of W-circle 1 the strongest section of the fire is attacked first in such a fashion that fire development is sometimes halted. The second is related to wind characteristics as some strategies can only be implemented under certain conditions: the Spot fire strategy can only be implemented during some VT trials: if the wind strength is under 5 and the spot fire is detected early, it is possible to stop its development by issuing a single DW command in the initial cell.

Each of the four strategies found in this analysis have different properties: the Barrier strategy can halt the development of the fire by using the CF command in a structured and timely fashion; the Stop strategy can stop the development of the fire by focusing on attacking key fires with copters and weak fires with trucks; the NonBarrier strategy can deter fire development by issuing a reduced amount of CF commands whilst at the same time issuing DW commands; and the Follow strategy can reduce the amount of terrain destroyed by focussing on extinguishing smaller fires in close proximity to the strongest fires. A mixed strategy such as Barrier contains a rich variety of behaviours that can combat the fire in many different situations; this characteristic makes this strategy a good choice when the efficiency of appliances is reduced during the testing phase. The comparison of these strategies continues throughout the following section by presenting a variety of metrics.

As previously mentioned the Cañas et al. (2005) study found evidence of cognitive inflexibility in the CT programme participants. To be more specific: participants in the CT group who use strategies such as Barrier and NonBarrier continue to use these strategies when there is a change in wind direction, even though this renders them ineffective, and participants who use Stop and Follow strategies continue to use these when there is a reduction in the efficiency of appliances when they are similarly ineffective. The most salient (negative) impact on performance occurs when participants do not apply a CF-DW strategy when the efficiency of appliances is decreased. In the VT group there is less inflexibility and participants switch strategies sooner.

## 3.4 Statistics and metrics

In the Cañas et al. (2005) study there were 80 participants but some of the data is missing or incomplete therefore the statistics presented here consider 72 participants. These statistics are centred on two variables: general performance and strategy use (section 2.3.1.5 describes how performance in measured).

### 3.4.1 Frequency of strategy use

Strategy use refers to the strategies used during a trial (a problem solver can switch strategies in the same trial). Table 3.3 shows the frequency of each version of the strategies in the training and testing phases. Barrier and NonBarrier strategies are used more frequently during CT than during VT. In the C trial (see table 3.2), with a wind strength of 6, the Barrier strategy is a good option because it may stop the fire which is developing quite quickly and soon reaches an intensity that surpasses the capability of the fire-fighting units. The NonBarrier strategy can also have good results if the most advanced fires are stopped and the remaining ones are attacked with water. The Stop strategy can work if key fires are attacked promptly, but the empirical data shows that only two participants were able to execute the Stop strategy with success in the CT trials.

Variable trials that start with low wind strength (trials 2, 3, 5, 6, 8, 10, 12 and 14) favour the use of the Stop strategy. In these trials a quick execution of Stop usually may result in a performance of 100%. The Barrier strategy consumes time for creating the barrier allowing the destruction of a few cells. In trials 2, 5 and 10, if the problem solver is able to extinguish the fire before the wind increases its strength, there are no complications. If the initial wind strength is above 5 (trials 1, 4, 7, 9, 11, 13 and 15) the Barrier or the Stop strategies seem to be the best option, but Barrier should be preferred because fire intensity may overcome appliance efficiency at any time. When implementing the Barrier strategy in a trial where the fires are in close proximity (e.g. trial 13) it's a good practice to create a single barrier for both fires. When there is a significant reduction in wind strength (trial 11) it is a good option to use DW commands only. In trials where wind strength is high and one of the initial fires is close to the eastern edge (trials 7 and 9), a good option is to use the Follow strategy in order to minimise the damage until it reaches the natural barrier imposed by the edge of the landscape. In trials where the initial fires are wide apart, such as trial 4, it is advisable to attack each fire with different units. If there are two mature fires in the trial (trial 4) the one that is furthest away from the eastern edge of the landscape should be selected (because if the wind is blowing towards the east this fire is the one that can destroy the most landscape). If the trial has high wind strength, a mature fire and a spot fire (trial 4) it is advisable to extinguish the spot fire with a single DW command and block the development of the remaining fire with a barrier. When the wind strength is high, it is common for the spot fire to expand rapidly early in the trial making it impossible to execute the Quick-spot-fire strategy. In the last variable training trial the initial wind strength is 5; in this trial Stop is the best option because the majority of fires can be extinguished effectively by DW commands. As can be seen there are several cues that the problem solver can use in order to select the appropriate strategy for a specific FireChief trial.

| Strategy | Version | Overall | CT | VT | CTW | VTW | CTE | VTE |
|---|---|---|---|---|---|---|---|---|
| Barrier | Line | 96 | 27 | 29 | 8 | 4 | 13 | 15 |
| | Semicircle 1 | 99 | 40 | 9 | 6 | 1 | 21 | 22 |
| | Semicircle 1A | 49 | 17 | 22 | 0 | 5 | 4 | 1 |
| | Semicircle 2 | 91 | 35 | 25 | 8 | 6 | 7 | 10 |
| | Semicircle 3 | 20 | 6 | 1 | 4 | 5 | 4 | 0 |
| | Circle 1 | 42 | 6 | 10 | 9 | 5 | 3 | 9 |
| | Circle 2 | 7 | 2 | 2 | 3 | 0 | 0 | 0 |
| | Diagonal | 13 | 5 | 4 | 1 | 0 | 2 | 1 |
| | Lines | 31 | 16 | 4 | 2 | 0 | 6 | 3 |
| | **Subtotal** | **448** | **154** | **106** | **41** | **26** | **60** | **61** |
| NonBarrier | MixBarrier | 78 | 34 | 19 | 7 | 4 | 9 | 5 |
| | SemiOrdered 1 | 52 | 13 | 23 | 4 | 7 | 3 | 2 |
| | SemiOrdered 2 | 15 | 8 | 3 | 3 | 0 | 0 | 1 |
| | FollowMix | 152 | 68 | 39 | 15 | 8 | 10 | 12 |
| | **Subtotal** | **297** | **123** | **84** | **29** | **19** | **22** | **20** |
| Stop | W-Circle | 334 | 52 | 207 | 14 | 60 | 0 | 1 |
| | W-Semicircle | 181 | 27 | 57 | 31 | 16 | 18 | 32 |
| | QuickSpotFire | 171 | 0 | 170 | 0 | 0 | 0 | 1 |
| | W-Encircle | 36 | 5 | 15 | 2 | 2 | 3 | 9 |
| | **Subtotal** | **722** | **84** | **449** | **47** | **78** | **21** | **43** |
| Follow | W-Follow 1 | 199 | 95 | 34 | 18 | 18 | 22 | 12 |
| | W-Follow 2 | 359 | 161 | 79 | 14 | 26 | 39 | 40 |
| | W-FollowSide | 51 | 21 | 22 | 0 | 1 | 2 | 5 |
| | **Subtotal** | **609** | **277** | **135** | **32** | **45** | **63** | **57** |
| | **Total** | **2076** | **638** | **774** | **149** | **168** | **166** | **181** |

**Table 3.3: Frequency of strategy use**

In the testing phase the Barrier strategy is used more often when appliance efficiency changes and the Stop strategy is used more often when the wind changes direction. Stop is used with less frequency when appliance efficiency is reduced, although it is used more frequently by VT participants. As previously mentioned, Barrier and Stop are more structured in comparison to

NonBarrier and Follow, and between Barrier and Stop the former is more complex because it involves more types of commands. For these reasons Barrier is more effortful and error prone and requires more practice (cf. Schunn & Reder, 2001).

### 3.4.2 Focusing the analysis

The goal of the FireChief task is to stop the fire using all commands, participants that do not seem to pursue this goal were not considered for the cognitive modelling stage. The main reason is that the cognitive model pursues the ultimate goal of stopping the fire and will act according to this goal. The practice of leaving out participants has been adopted in other studies. Schunn & Reder (2001) encountered similar difficulties in accounting for the full range of performance data and they reduced the number of participants modelled (34 of 57) because they lacked the data for analysing the measure of strategy adaptivity that they were studying. The selected subgroup is comprised by participants that exhibit engagement with the task, that is, participants that demonstrate the intention of stopping the fire. Modelling behaviours not directly related to *stopping the fire* escapes the aims of this research. Because of the lack of other behavioural data, such as eye-movement data or verbal protocols, it is hard (if not impossible) to determine a participant's intentions during the lapses in which they are not issuing commands and hence extracting meaningful elements for modelling is not possible.

| Transition | Barrier | NonBarrier | Stop | Follow |
|---|---|---|---|---|
| M-M | 15.15 | 14.85 | 20.06 | 13.45 |
| DW-M | 26.13 | 25.81 | 38.88 | 31.82 |
| CF-M | 16.82 | 11.69 | 2.97 | 5.18 |
| M-DW | 27.10 | 26.94 | 39.13 | 32.42 |
| DW-DW | 6.71 | 8.60 | 4.28 | 11.87 |
| CF-DW | 0.89 | 0.54 | 0.72 | 0.34 |
| M-CF | 15.88 | 10.46 | 2.69 | 4.74 |
| DW-CF | 1.81 | 1.67 | 1.00 | 0.89 |
| CF-CF | 3.36 | 2.56 | 0.69 | 0.66 |

| Transition | Barrier | NonBarrier | Stop | Follow |
|---|---|---|---|---|
| M-M | 15.63 | 16.01 | 15.33 | 13.77 |
| DW-M | 23.56 | 22.69 | 22.48 | 31.84 |
| CF-M | 17.20 | 12.79 | 9.42 | 4.45 |
| M-DW | 24.52 | 23.86 | 22.88 | 32.39 |
| DW-DW | 5.75 | 7.12 | 3.45 | 10.27 |
| CF-DW | 0.81 | 0.73 | 0.45 | 0.41 |
| M-CF | 16.25 | 11.62 | 8.92 | 4.13 |
| DW-CF | 1.76 | 1.74 | 1.02 | 0.81 |
| CF-CF | 3.13 | 2.33 | 1.73 | 0.64 |

**Table 3.4: Normative transition matrices for the CT (top) and the VT (bottom) conditions (M = Move, DW = Drop Water, CF = Control Fire. M-M indicates how many times a Move was followed by another Move)**

Participants' engagement is measured in terms of trial performance and pattern of command execution. In order to reduce the subjectivity during the selection process considering the pattern of command execution transition matrices were used. During the first stage transition matrices that represent the patterns of command use for each of the four main strategies depicted in the third level of figure 3.6 were created for both training programmes. These matrices were extracted by averaging the individual transitions between commands for every trial classified under the different strategies. Each matrix has 9 rows due to the permutation of the 3 FireChief commands. Table 3.4 shows the resulting normative transition matrices. Barrier and NonBarrier strategies use more CF commands than the Stop and Follow strategies and thus generate more CF-related transitions. Similarly Stop and Follow strategies use more DW commands. The Barrier strategy uses more CF commands than the NonBarrier strategy particularly before and after a Move. In the CT trials Stop uses more DW commands than Follow and there are more M-M transitions. In the VT trials Stop uses fewer DW commands. The less structured strategies, NonBarrier and Follow, use one DW followed by another DW many more times.

During a second stage transition matrices are converted into vectors and a (stepwise) multiple regression analysis is performed with the empirical transition matrix (of each participant) as the dependent variable and the normative transition matrices of each of the four strategies as the predictors. This process distinguishes between training and test trials. After the results are obtained those participants whose predictors suggest engagement with the tasks are considered as cases for modelling. Figure 3.7 shows the results of this analysis considering the CT and CTW condition. The first column enumerates engaged participants. The second and fourth columns show which strategies out of the 4 normative strategies are the best predictor (the first strategy in the list is the one that has the highest predictive power) of the strategies executed by the participant during the training and testing phase respectively. The third and fifth columns show the value of the R coefficient for both the training and testing phases.

| Subgroup | Phase | | | | Rest | Phase | | | |
|---|---|---|---|---|---|---|---|---|---|
| CTW | Training | | Testing | | CTW | Training | | Testing | |
| Participant | Predictors | R | Predictors | R | Participant | Predictors | R | Predictors | R |
| 1 | NB, ST | 0.995 | BA, ST | 0.999 | 1 | FL | 0.96 | ST | 0.918 |
| 2 | BA | 0.953 | BA, FL | 0.978 | 2 | FL | 0.984 | FL | 0.98 |
| 3 | BA, ST, FL | 0.997 | BA, ST | 0.989 | 3 | FL, NB | 0.996 | FL | 0.983 |
| 4 | BA | 0.971 | BA, NB | 0.991 | 4 | FL | 0.986 | ST | 0.994 |
| 5 | FL | 0.969 | FL | 0.984 | 5 | FL | 0.939 | ST, FL | 0.996 |
| 6 | BA, NB | 0.996 | ST, NB | 0.997 | 6 | FL | 0.999 | NB, FL | 0.999 |
| 7 | BA | 0.994 | NB, ST | 0.994 | 7 | FL, NB | 0.999 | ST, NB | 0.984 |
| 8 | BA | 0.983 | NB | 0.952 | 8 | FL | 0.989 | FL, BA, ST | 0.993 |

**Table 3.5: Strategy use considering transition matrices CTW group**

Table 3.5 shows that the (selected) "Subgroup" frequently uses the Barrier strategy during training, even though a change in the wind direction makes the execution of the Barrier strategy harder participants in this group do not tend to switch to another strategy during testing. This is evidence of cognitive inflexibility. In the "Rest" group the majority of participants use the Follow strategy during the training phase, and they show low performance in most of the trials. When the wind changes direction the "Rest" Participants in the 'Rest' group show low performance in most of the trials, execute a low number of commands and do not adapt to the environment by changing their pattern of command use.

Similar results were obtained for the other three experimental groups. The results are not presented here as they mirror the data presented in table 3.5 where participants in the Rest group are best predicted by the Follow (FL) strategy with low use of commands. What is salient is that the selected subgroup uses CF commands during training and during testing the majority of participants stick to the Barrier strategy which is the appropriate adaptation as there is no way of stopping the fire without using CF commands when the efficiency is reduced because strong fires cannot be stopped so they burn almost freely. In the VTW condition the "Subgroup" shows good performance during testing and many of its members changed strategies compared to the ones used during training. In the VTE condition the "Subgroup" uses a mixture of strategies during training and start using CF commands and attempt to create barriers for stopping the fire during the testing phase which is a convenient adaptation. In summary, the selected cases for modelling is comprised by those participants that show a pattern of command use other than an infrequent use of DW commands in both the training and test conditions.

#### 3.4.2.1 The limitations of transition matrices

Table 3.6 shows the averaged matrix of transitions for the strategies used more frequently for the CT condition considering the selected "Subgroup". The transition matrices of Barrier Line and Barrier Lines strategies are quite similar although the spatial distribution of commands is not (see the corresponding images of the strategies in section 3.3.1.1 for the Line strategy and in section 3.3.1.2 for the Lines strategy). The lack of consideration of spatial information is the major drawback of transition matrices. As a consequence transition matrices are not able to capture, for example, differences in the form of the barrier. Another example: consider the similarity in the number of M-DW and DW-M transitions for strategies W-Circle and W-Follow-1. Although the number of these transitions is very similar, the essential difference between these strategies is how commands are distributed in the landscape.

| Transition | Semicircle 1 | Semicircle 2 | Line | Lines | Follow Mix | Mix Barrier | W-Circle | W-Follow 1 | W-Follow 2 |
|---|---|---|---|---|---|---|---|---|---|
| M-M | 14.89 | 13.04 | 14.67 | 15.62 | 14.63 | 15.28 | 20.13 | 16.70 | 11.19 |
| DW-M | 29.47 | 17.61 | 27.39 | 25.08 | 23.71 | 27.00 | 38.37 | 38.00 | 29.38 |
| CF-M | 15.16 | 18.70 | 17.83 | 17.15 | 9.96 | 12.56 | 2.90 | 4.80 | 3.05 |
| M-DW | 30.82 | 18.61 | 28.44 | 26.23 | 24.58 | 28.78 | 38.53 | 38.10 | 29.95 |
| DW-DW | 4.97 | 4.35 | 6.17 | 10.23 | 10.04 | 7.89 | 4.23 | 10.60 | 13.19 |
| CF-DW | 0.89 | 1.00 | 0.67 | 0.46 | 0.71 | 0.39 | 0.73 | 0.20 | 0.24 |
| M-CF | 13.82 | 17.74 | 16.72 | 16.23 | 9.00 | 10.67 | 2.67 | 4.50 | 2.67 |
| DW-CF | 2.29 | 1.83 | 1.78 | 1.46 | 1.54 | 2.22 | 0.97 | 0.70 | 0.67 |
| CF-CF | 6.39 | 1.96 | 1.61 | 2.77 | 2.92 | 2.56 | 0.63 | 0.60 | 0.33 |
| Performance | 89.77 | 79.35 | 79.02 | 73.16 | 65.24 | 83.15 | 95.16 | 63.53 | 59.43 |

**Table 3.6: Transition matrices for strategies in the CT**

Table 3.7 shows the transition matrices for the VT condition. Note that there is a difference in the number of M-DW and DW-M transitions for the W-Follow 1 and W-follow 2 strategies in comparison with the CT condition. Due to the variable complexity of VT trials the amount of DW commands required to successfully execute the Stop strategy tends to decrease. Nevertheless, once again, transition matrices do not distinguish between the W-Follow 1 and

W-Follow 2 strategies because they cannot capture the higher dispersion of commands characteristic of W-Follow 2.

The second limitation of transition matrices is the lack of consideration of the temporal distribution of commands. Consider the M-CF transitions of strategies Semicircle 1, Semicircle 2 and Line in table 3.7. This data is telling us that participants tend to issue CF commands after executing a Move command; if this sequence is executed by the same unit this is evidence of 'waiting' behaviour (that is, a participant must wait until the unit arrives at the desired location before issuing the CF); nevertheless it is not possible to assume that this sequence is executed by the same unit nor is it possible to know how much time has elapsed between the execution of the Move and the CF command.

| Transition | Semicircle 1 | Semicircle 2 | Line | W-Scirc | Follow Mix | Mix Barrier | W-Circle | W-Follow 1 | W-Follow 2 |
|---|---|---|---|---|---|---|---|---|---|
| M-M | 14.70 | 17.20 | 17.75 | 17.33 | 17.08 | 14.33 | 17.11 | 12.75 | 14.54 |
| DW-M | 20.70 | 11.20 | 15.75 | 41.00 | 23.23 | 16.50 | 24.00 | 31.00 | 30.15 |
| CF-M | 18.50 | 19.00 | 20.50 | 1.67 | 12.23 | 13.50 | 7.21 | 2.50 | 3.15 |
| M-DW | 21.80 | 13.20 | 15.50 | 40.67 | 24.62 | 18.00 | 24.11 | 31.75 | 30.69 |
| DW-DW | 3.80 | 1.20 | 5.25 | 3.33 | 6.38 | 2.17 | 5.37 | 7.25 | 9.62 |
| CF-DW | 0.60 | 1.60 | 0.63 | 1.00 | 0.69 | 1.17 | 0.42 | 0.00 | 0.54 |
| M-CF | 17.50 | 17.00 | 20.25 | 2.00 | 11.08 | 12.00 | 6.79 | 2.00 | 3.00 |
| DW-CF | 1.70 | 3.80 | 0.63 | 0.33 | 1.54 | 2.83 | 0.84 | 0.50 | 0.77 |
| CF-CF | 1.50 | 4.00 | 3.13 | 0.67 | 0.85 | 1.83 | 2.63 | 0.50 | 0.54 |
| Performance | 87.79 | 68.69 | 61.87 | 62.90 | 48.42 | 66.01 | 96.16 | 56.8075 | 53.13 |

**Table 3.7: Transition matrices for strategies in the VT**

### 3.4.3 Subcultures

An analysis of subcultures was conducted with the aim of identifying patterns of strategy use among participants. Subcultures are a means of labelling individual differences in predisposition for using certain types of strategies under particular circumstances. A subculture is defined as the set of participants who demonstrate a particular style of strategy use. The process of finding subcultures started by identifying the distribution of strategy use for each participant making a distinction between the training and testing phases. During the next step every participant was characterized in terms of dominant strategies. In most of the cases a single strategy was used more that 50% of the time, a phenomenon confirmed by transition matrices (see table 3.5). The last step was to group participants based on their dominant strategies. The result of this process is presented in table 3.4. Subcultures 1, 4, 6 and 11 are the most frequent whilst subcultures and 1, 3, 8 and 10 are the most successful. The CT group is mainly distributed among subcultures 1, 2, 4 and 6. Problem solvers on the VT programme are distributed in subcultures 4, 10 and particularly 11. Subcultures 10 and 11 made use of a version of the Stop strategy called the Quick-Spot-Fire strategy that can be implemented only under the VT programme. The majority of participants in group CTW belong to subcultures 1, 3 and 4 and therefore execute structured strategies more often. The majority of participant in the CTE group belong to subcultures 1, 6 and 8. The frequency of subculture 8 is expected because a reduction in the efficiency of appliances makes the use of CF commands necessary in order to obtain good performance. The majority of participants in group VTW belong to subcultures 3 and 4 which are predominated by the Stop strategy. The majority of participants in the VTE group belong to subcultures 1, 4 and 6. As in the case of the CTE group the use of the Barrier strategy increases as a response to the appliance efficiency reduction.

| Subculture | Description | CT | VT | CTW | CTE | VTW | VTE |
|---|---|---|---|---|---|---|---|
| **1** | Barrier toNonBarrier or Follow | 6 | 1 | 3 | 4 | 1 | 6 |
| **2** | NonBarrier to Follow | 4 | 0 | 0 | 0 | 0 | 1 |
| **3** | Stop | 2 | 0 | 3 | 0 | 5 | 0 |
| **4** | Stop to NonBarrier or Follow | 4 | 6 | 3 | 0 | 4 | 4 |
| **5** | Decrement-structure-Follow | 3 | 0 | 0 | 2 | 0 | 0 |
| **6** | Increment-structure-Follow | 9 | 1 | 1 | 4 | 1 | 3 |
| **7** | Follow | 3 | 0 | 1 | 1 | 1 | 1 |
| **8** | Barrier | 0 | 1 | 0 | 4 | 2 | 2 |
| **9** | Stop to Follow or NonBarrier | 0 | 1 | 0 | 0 | 2 | 0 |
| **10** | Stop (spot fire) to NonBarrier | 0 | 6 | 0 | 0 | 0 | 0 |
| **11** | Stop (spot fire) to Follow | 0 | 15 | 0 | 0 | 0 | 0 |
| **12** | Follow to NonBarrier | 0 | 0 | 2 | 0 | 0 | 0 |

**Table 3.8: Distribution of participants in subcultures**

Up to this point, the analysis of the data has produced three outcomes: firstly a hierarchy of strategies was identified. Second, the effectiveness of the different strategies in relation to types of scenarios was also identified. Third, the emergence of subcultures was observed. These findings pave the way for understanding the effect of training condition on (1) use and consolidation of strategies and the responses seen, both in kind (what the changes are and their effectiveness) based on individual preference and what the training condition trained participants for and (2) propensity to change strategies in response to wind or efficiency changes combined with the effectiveness of the preferred strategies in the new conditions.

### 3.4.4 Performance by trials

The mean performance of the 72 participants is 71.9. Trial complexity was taken into account: the base performance (section 3.1.1.2) of a trial is used to perform min-max normalization. Table 3.9 shows the average performance of all participants and the selected "Subgroup". When considering all participants, the CTE and VTE groups have lower performance than groups for CTW and VTW during testing. The selected subgroup has a different pattern, participants in groups CTW and CTE have equivalent performance levels during testing. This phenomenon can be explained by resorting to strategy use (discussed in later chapters).

| | Training | | Testing | |
|---|---|---|---|---|
| Group | Overall | Subgroup | Overall | Subgroup |
| CTW | 68.90 | 76.55 | 74.39 | 78.81 |
| CTE | 71.14 | 80.62 | 65.99 | 78.70 |
| VTW | 75.56 | 76.63 | 77.31 | 82.12 |
| VTE | 76.70 | 79.27 | 63.38 | 76.13 |

**Table 3.9: Average performance by conditions**

Figure 3.7 shows the overall performance (CT and VT) and the performance of the selected-subgroup (CT' and VT') by trials during the training phase. One-tailed one-way between-subjects ANOVAs are used in this section. The average performance of participants in the "Subgroup" is significantly higher than of the "Rest" group. (F(80, 1) = 7.45, p<0.05).

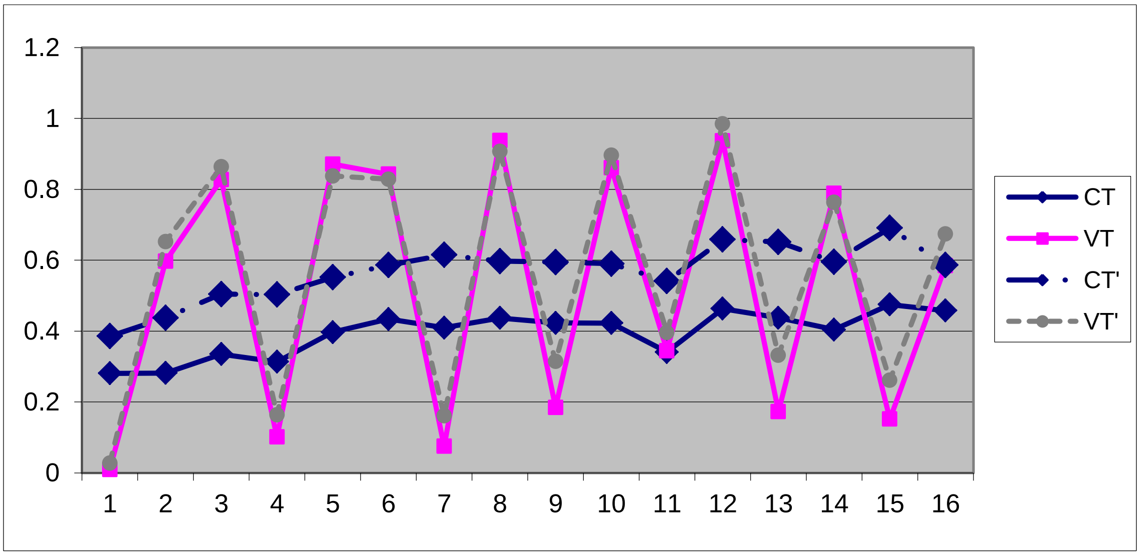



**Figure 3.7: Performance (Y-axis) by trials (X-axis) during the training phase**

To assess the impact of practice in the CT group normalized performance in the first four trials was compared with performance in the last four trials. There is a significant effect of practice in the "Subgroup" (F(1,32)=5.175, p<.05). The "Rest" group does not show a performance improvement (F(1, 34)=0.27, p=.60). Another significant interaction was found between task complexity of VT trails and performance (F(2,42)=45.83, P<.001). This result supports the view that complexity in FireChief is heavily related to wind conditions. VT participants show considerable performance variability during the whole training programme.

Figures 3.8 (a) and (b) show the average performance for the testing phase for all experimental groups. Performance is better for groups CTW and VTW in comparison with groups CTE and VTE (F(1,76) = 6.44, p<.05). This result suggests that a change in appliance efficiency was more demanding for problem solvers than a change in wind direction. There is no significant difference between groups CTW' and VTW' in comparison with groups CTE' and VTE' (F(1,34)=0.06, p=.80). This means that the selected subgroup is able to deal with the complexity presented by the reduced efficiency condition. The difference between subgroups CTE' and VTE' compared to groups CTE and VTE is larger compared to the CTW condition. The reason behind this significant difference is traced to the use of strategies in the following chapters.

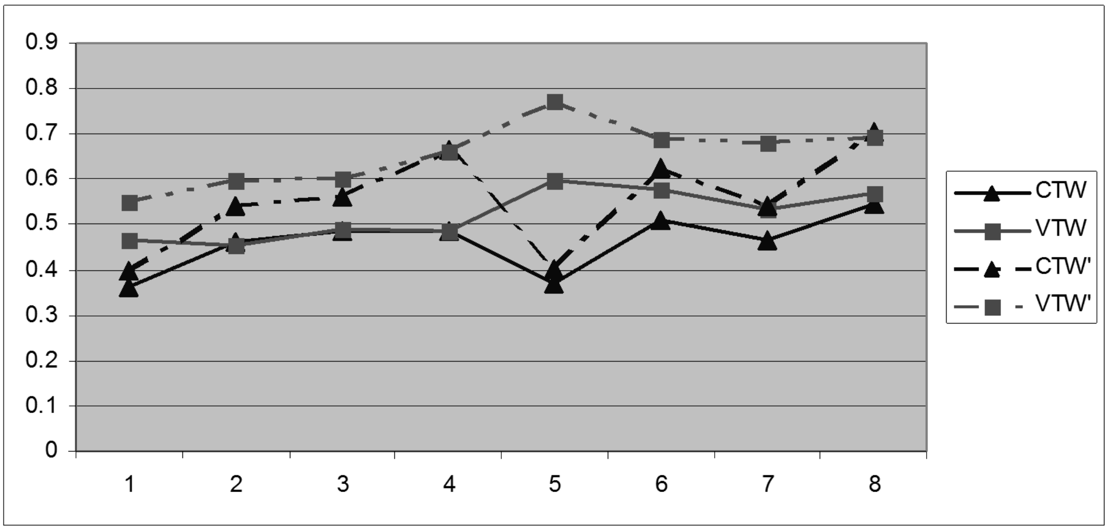



**Figure 3.8 (a): Performance (Y-axis) by trials (X-axis) for testing groups CTW and VTW**

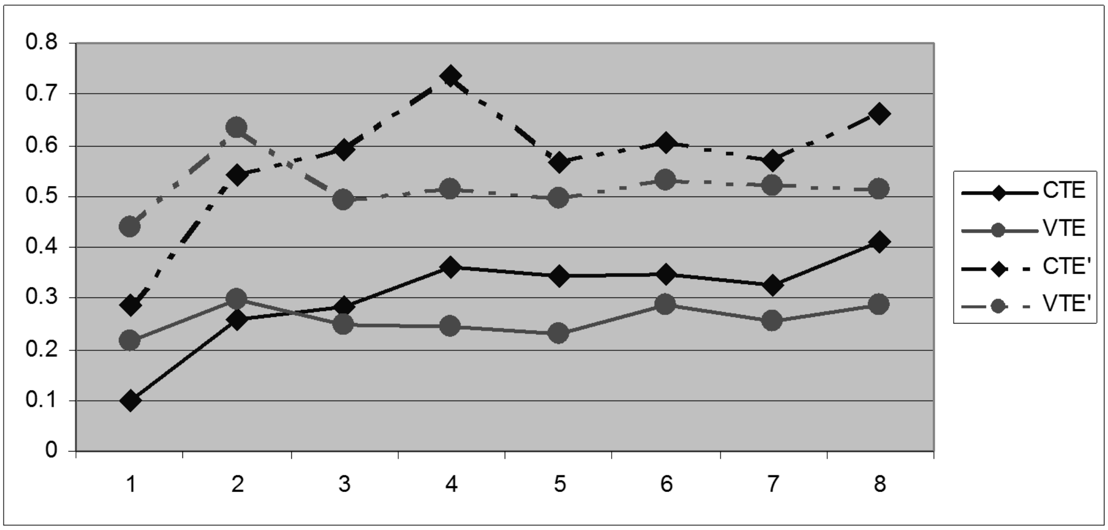



**Figure 3.8 (b): Performance (Y-axis) by trials (X-axis) for testing groups CTE and VTE**

Overall the selected subgroup exhibits the patterns described in the Cañas et al. study (2005): (1) there is more strategy variability in participants of the VT group compared to participants of the CT group, (2) VT facilitates better performance when the wind changes direction and (3) there are significant differences between strategies on various performance measures including the amount of landscape saved from the fire and the pattern of unit use. The following sections continue with the analysis of strategy use for the selected "Subgroup" only. The main reason behind this decision is that participants in the 'Rest' subgroup persist in using an unsuccessful version of the Follow strategy and therefore do not show strategy variation nor adapt to the environmental change that occurs during the testing phase. Relevant statistics related to strategy use are presented in section 6.2.2 where data generated by the model can be compared. The aim of the analysis in the following section is twofold: to understand more about how participants select and execute commands whilst achieving the appropriate level of granularity to construct and evaluate a cognitive model. In this sense the modelling work is used to understand the relationships evidenced by these measures.

### 3.4.5 Command use metrics

Strategy execution depends upon how commands are issued. PAT (section 3.2.2) processes the protocols to obtain the measures of performance presented in these sections. The first set of metrics is based on the frequency and latency of commands. In gathering these metrics it is considered that the time elapsed between the timestamp associated with a command minus the timestamp associated with the previous command represents the time taken to execute the command. Another measure related to command use corresponds to the time elapsed between commands executed by the same unit. This measure is particularly useful for identifying repeated or cancelled commands. It is important to identify incomplete commands for many reasons: they can show us how many times a participant issued a DW command over a fire that exceeded the capability of a unit, how many times the fire cancelled the execution of a CF command or how many times participants repeated a command.

As mentioned, given the spatio-temporal nature of FireChief, it makes sense to add spatial and temporal information to the analysis of commands. A set of metrics were extracted from participants' protocols which incorporate these two dimensions. It is well known that motor actions have longer execution times than perceptual or cognitive actions. This fact makes the spatial distribution of commands quite important when comparing strategies. Assuming the same cognitive and perceptual actions, a group of commands issued within a smaller area will take less time than a group of commands issued over a larger area. The latter group requires more time because the mouse pointer must be moved longer distances. There are other factors besides the length of mouse movements, such as the complexity of the trial, which influence the duration of commands.

#### 3.4.5.1 *Move* command metrics

The maximum length of a FireChief movement is around 30 cells (this happens when a unit moves from one extreme in the landscape to the other). Move commands are classified according to their length, where length is defined as the Euclidean distance between the extremes of the movement. Figure 3.9 shows the distribution of Move commands in relation to the length of the movements. The x-axis shows the different Move lengths and the y-axis shows their frequency. The shortest movements are 1 cell long whilst the longest movements are longer than 10. The Move command provides more information about differences in strategy use compared to other commands because it usually serves as a step in executing the other commands. Figure 3.9 shows that there is a preference for short movements (sizes 1 and 2) for all strategies. A one-tailed one-way within-subjects ANOVA revealed that there is also a significant difference between strategies Barrier and Stop compared to NonBarrier and Follow considering movements of length equal to 1 ($F(1,262)=29.670$ $p<.001$). The reason is that strategies Barrier and Stop are more structured than Follow and NonBarrier and therefore the frequency with which commands are issued in close proximity to each other is higher.

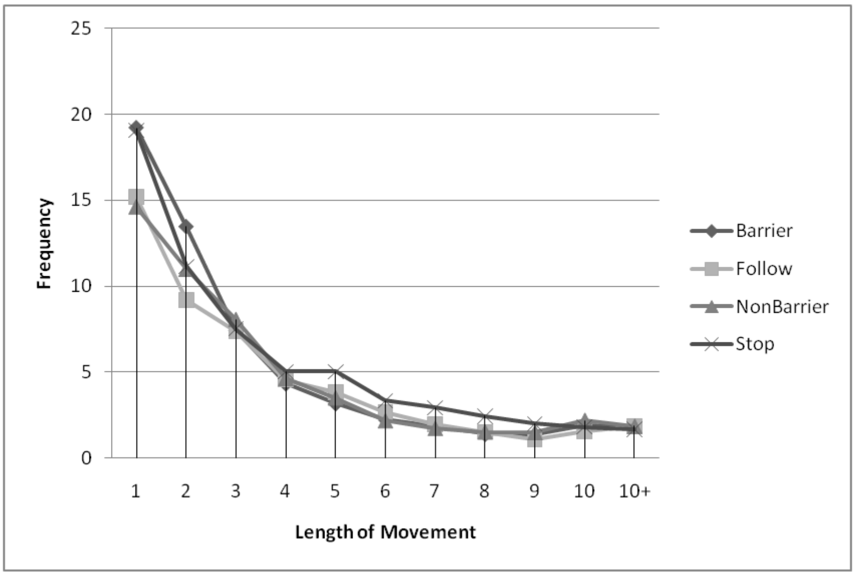



**Figure 3.9: Frequency of Move commands by distance traversed in the CT group**

Figure 3.10 shows the results of the movement length analysis for the VT group. In this case the Follow strategy has more movements of length equal to 1 than any other strategy. Presumably this is a consequence of some trials in which the wind strength is reduced when much terrain is on fire, allowing for an implementation of Follow in which units are moved short distances. Note also that the number of short movements executed by the Stop strategy is the lowest of all the strategies. This is a consequence of the number of trials in which Stop is used where low wind conditions allow for a quick extinction of the fire.

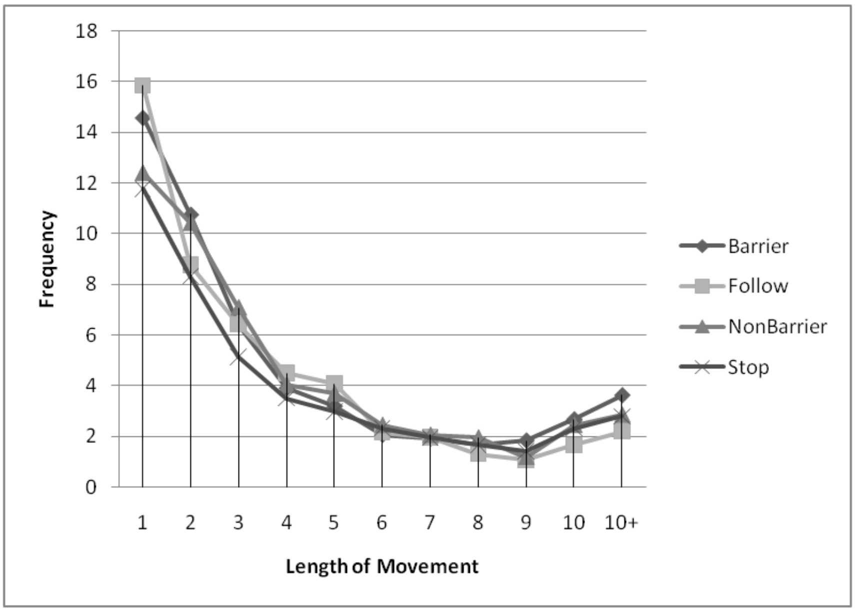



**Figure 3.10: Frequency of Move commands by distance traversed in the VT group**

### 3.4.5.2 *Control Fire* command metrics

CF commands can be of two types: those that are part of a barrier and those that are not. Considering all trials a one-tailed one-way within-subjects ANOVA revealed that the execution of a CF command involved in forming a barrier takes significantly less time than a CF command not forming part of a barrier ($F(1,852)=167.5$, $p<.001$). To determine whether or not a CF command belongs to a barrier a function that combines spatial and temporal information is used. A CF command is considered part of a barrier if its Euclidian distance to any other cell with a CF command is equal or less than 1.5 and there are fewer than 100 generations between the CF commands. This function is implemented in PAT.

#### 3.4.5.3 *Drop Water* command metrics

There is a significant difference between using trucks and copters to execute DW commands: a one-tailed one-way within-subjects ANOVA revealed that DW commands issued by copters take less time than DW commands issued by trucks ($F(1,1436)=101.13$ $p<.001$).

#### 3.4.5.4 The *Move* command and the creation of barriers

Results reveal that there is an interaction between how the Move command is used and the duration of this command (the intention of a movement is determined by the action executed next). All Move commands was classified under these categories: position a truck that will issue a CF command that will contribute to the structure of a barrier, position a truck that will issue a CF command that will not contribute to the structure of a barrier, position a copter, refill a unit, or precede another movement. On average participants spent 13.12 generations executing a Move command which will be followed by a CF command close to one or more completed CF commands (this is an operational definition of a CF that belongs to a barrier of CFs). In contrast, participants spend 16.57 generations when executing a Move command which will be followed by a CF command that does not belong to a barrier. A one-tailed one-way within-subjects ANOVA revealed that there is a significant difference between these two types of movements ($F(118,1)=30.85$ $p<.001$). More cognitive processing is required to place a CF command that is part of a barrier than when it is not, and this excess of cognitive processing should represent a higher cost in time. Chapter 4 (section 5.2.3.1) describes how these cognitive processes were modelled.

## 3.5 Modelling problem solving behaviour

This chapter describes the results obtained by conducting a new analysis of the Cañas et al. (2005) data. Identifying strategies in a complex dynamic task such as FireChief required methods for dealing with the temporal and spatial dimensions of the data. The approach followed in this analysis was to use a tool that enabled the discovery of these temporal and spatial patterns of commands.

A topic that emerges from this analysis is how a strategy is implemented, that is, in order to obtain good performance an adequate strategy selection is not enough as an adequate execution of the chosen strategy is required too. An adequate implementation of a strategy is based on the correct execution of the individual commands that make up the strategy. Constructs such as strategy consolidation and the use of bottom-up or top-down control play a role in how strategies are implemented. The statistics and metrics presented here show that participants execute commands with significant differences according to training programme and strategy selection. The cognitive model is particularly useful in exploring these topics. An

ACT-R model, in addition to a pattern of commands, provides plausible perceptual and cognitive operations with explicit latencies making it possible to compare these measures with participants' data. There are many actions that lie behind the execution of each command registered in the protocols that are not explicated or explained. These actions include cognitive actions such as deciding what to do, perceptual actions such as looking for a unit in the landscape or manual actions such as moving the mouse pointer. The cognitive model fills in these gaps to reveal the complex problem solving processes behind the issuing of commands and by doing so provides a description of how strategies are executed at a finer grain of detail.

Moreover there are interesting interactions found in the data that can be further investigated using the cognitive model.

I. There is more strategy variability in participants of the VT group compared to participants of the CT group. This variability can be explained due to the ever-changing characteristics of the VT trials that ultimately affect trial complexity and to the opportunity that the CT condition provides for the consolidation of strategies. In this respect the cognitive model can show how the different strategies are affected by changes in the complexity of trials and can provide a description of how strategy consolidation may occur.

II. VT facilitates better performance when the wind changes direction. The group with the highest performance is VTW where every member improved their performance when the wind changed direction. The CTW group was slightly affected by a change in the wind direction and experienced a slight detriment in performance in this condition. This phenomenon can be explained due to the cognitive inflexibility of participants in the CT programme that have consolidated a strategy that is not the best choice for the wind-direction-change scenario. In this respect the model can explore two things: why VT participants are more prone to changing strategy as a product of their previous experiences with the task and how participants in the CT group have problems with environmental change.

III. Participants that used the Barrier strategy during CT deal much better with a reduction in the efficiency of appliances compared with other participants. In this case participants that have consolidated the Barrier strategy during CT are less affected by the reduction in the efficiency of DW commands. Besides showing the advantages of consolidating the Barrier strategy in this scenario, the model can provide an account of the adaptations in the use of copters when supporting the creation of the barrier.

IV. There are significant differences between strategies in various performance measures including the amount of landscape saved from the fire, the pattern of unit use, and the latency and use of commands. For example, it is hypothesized that the Barrier strategy is more difficult (i.e. requires more cognitive and perceptual operations) in comparison with the other strategies, and that a certain degree of consistency in task characteristics is necessary in order to successfully execute it. The cognitive model can shed light in this respect because the effort required by a strategy is related to the number of productions a strategy needs to fire.

## 3.6 Summary

In conjunction with chapter 2, where a set of constraints has been extracted from the CPS theory (i.e. the CPS paradigm, determinants of CPS performance, task characteristics and cognitive demands) and the nature of the FireChief task, this chapter adds a set of constraints from two sources: the particular configuration of the training and test programmes (sections 3.1.1 and 3.1.3 respectively) and a well-defined set of strategic patterns (section 3.3). These constraints are used in the development of the cognitive model. The next chapter describes the implementation of the cognitive model. The data offers a rich set of interactions that at this point can be traced to various theoretical constructs such as cognitive inflexibility, consolidation of strategies, strategy selection, time pressure, complexity, the dynamic nature of the environment, and feedback, among others. The objective of creating a cognitive model is to lay down these constructs in the form of well-defined computational representations (i.e. ACT-R chunks and productions) in order to test hypotheses and obtain explanations related to the interactions observed in the data. Because the model is implemented using the ACT-R cognitive architecture an additional set of constraints is placed upon the development of the model: the behaviour of participants must be modelled by means of pre-established ACT-R mechanisms. In this sense the model creates a bridge between the specification of CPS theoretical constructs and ACT-R theory elements.

# 4 The Cognitive Model of FireChief

This chapter provides a detailed description of the ACT-R cognitive model of FireChief behaviour. For this study strategies are considered the basic block of behaviour in the attempt to explain performance differences. For this reason the modelling effort was centred on strategy use and implementation, this decision impacted several aspects of the model such as what processes are included, how the knowledge was encoded and how procedural and declarative knowledge is used. The first section provides a high level description of the model that should serve as the basis for comprehending the rest of the chapter. The second section describes the implementation of the FireChief microworld in the LISP language; this version was developed to allow interaction between the model and the task. The most relevant ACT-R mechanisms are discussed next. The fourth section is related to the different functional levels in which FireChief behaviour is modelled, this is the largest section of this chapter where the modelling of FireChief strategies is presented. In the last section the various outputs of the model and how they are used for understanding more about problem solving behaviour are described.

## 4.1 An overview of the cognitive model

A couple of cognitive paradigms were leveraged for the construction of the model. The first paradigm is called *Competing Strategies (*Taatgen et al., 2006) where several strategies are implemented and then compete with each other in solving the problem task. In this paradigm, ACT-R's utility learning mechanism ensures that the strategy with the best probability of success and the lowest cost is used more often (see section 4.3.1). According to Taatgen et al. (2006) utility learning is a useful mechanism in tasks where there are multiple cognitive strategies, but where it is unclear which one is best. An important consideration here is how to implement these strategies in order to allow a fair competition, an issue that is explored in this research. The second paradigm is called *Perceptual and Motor Processes (Kieras & Meyer, 1997)*. In this paradigm the timing behaviour of the perceptual and motor systems is modelled in approximate form using formulas and standard parameters such as Fitts' law (Fitts, 1954). Because perceptual and motor processes are frequently used during the interaction of a problem solver with FireChief the importance of modelling such actions is considerable. Taatgen et al. (2006) argue that modelling motor and perceptual actions increases the quality of the model.

There are some principles that guide model behaviour. Brehmer & Dörner (1993) stress the importance of checking the development of important variables in microworlds, they being the temperature in Coldstore, the amount of destroyed cells in FireChief or the amount of ground water in Moro (section 2.3.1 describes all these microworlds). According to Quesada, Kintsch & Gomez (2002) relating feedback to the originating action (assigning blame to actions) can be complex. For example in FireChief it is hard for the problem solver to make sense of the feedback from dropping water over a cell on fire (and thus extinguishing it) while there are another twenty cells burning at the same time. Making sense of this feedback requires assigning a value to the particular action of dropping water. Dropping water over a cell requires a rather complex chain of cognitive, perceptual and motor actions that ultimately consume time, and there are multiple ways of combining these operations for obtaining the same result: dropping water over a cell. In order to improve performance participants need a way to learn the best way to execute actions. Another example is when time is wasted. A unit in movement is disabled for a period of time; if a participant decides to wait until it arrives at its destination and meanwhile some landscape is destroyed, how is this feedback to be processed by this participant? It may be the case that the benefit of waiting for this particular unit overshadows the cost of allowing the destruction of terrain. Although FireChief represents an ill-defined problem, if a clear-cut way of assessing the effectiveness of one's actions is established, it should be possible to guide behaviour. For this reason the adequate processing of feedback is paramount for controlling dynamic systems. How problem solvers process feedback from a dynamic environment (how they evaluate the effectiveness of their actions) is a focal topic of the research described in this document and is the central design consideration for the cognitive model. Atkin et al. (1999) defined some characteristics for high performance artificial controllers such as the ability to process sensor information, react to a changing environment in a timely manner, integrate reactive and cognitive processes to achieve abstract goals, and interleave planning and execution. In the context of ACT-R these abilities are realized through a combination of perceptual, motor and cognitive actions which are governed by a combination of top-down and bottom-up forms of control. Due to the dynamic nature of FireChief the state of the simulation itself is used as an external memory so there is no need to store complete information about the environment in memory, although some elements are kept in WM. The characteristics of FireChief require a monitoring procedure that renews the 'picture' available for the ACT-R's visual module at short intervals.

The application of these principles generated the basic workflow of the model shown in figure 4.1. Other authors have conceptualized cognition as a workflow such as Veksler, Gray & Schoelles (2007). This workflow iterates throughout the model's execution and it finishes with the awarding of a (positive or negative) reward. A Decision Point requires the selection of single option among different alternatives, Nellen & Lovett (2004) use the term Choice Point to refer to the same idea. The high branching factor at every *Decision Point* allows the emergence of several behaviours and also there are external events that can interrupt the flow of actions in the cycle: alarms and visible changes in the environment (explained later).

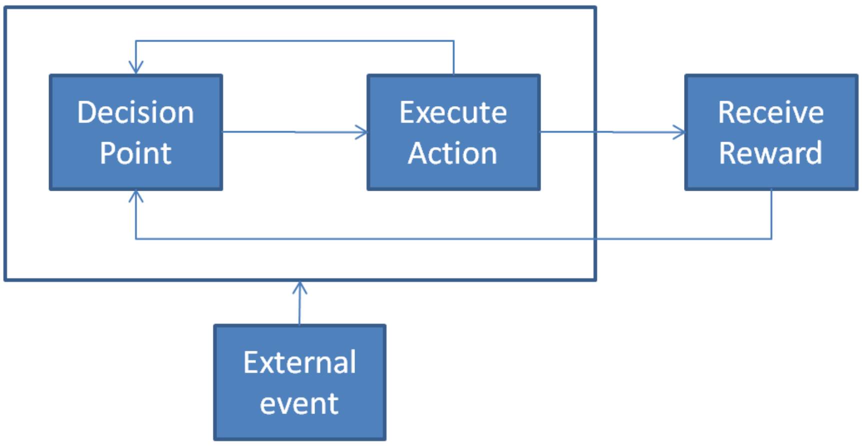



**Figure 4.1: The local workflow of the model**

In a similar fashion to a human participant, the model can use three kinds of commands for stopping the fire (Move, Drop Water and Control Fire) and in executing these commands it must adhere to the constraints imposed by the ACT-R architecture. The responsibility for presenting the trial, controlling the development of events, logging the model's behaviour and in general enabling the task lies with a LISP version of FireChief developed for this research. At the beginning of the trial, a large set of productions is available in the model's procedural memory; these productions are able to match the content of the different ACT-R buffers and execute actions (section 2.4.1.3). These rules will compete with each other during the running of the model because only one rule can be executed in each ACT-R cycle (section 2.4.1.3). This competition is based on utility; the initial utility of these productions is assigned randomly. The model needs to fire these rules for executing a diversity of cognitive, perceptual and manual actions which ultimately produce FireChief commands. The ultimate goal of the local workflow is to support a continuous competition of intentions mediated by the credit-assignment process described later.

### 4.1.1 Purpose of the model

The purpose of the cognitive model is to further explore topics related to the execution of strategies in complex dynamic tasks. Chapter 3 showed that there is significant interaction between strategy use and performance in the Cañas et al. (2005) data. In this respect the model offers a deeper understanding of this interaction by identifying the factors that explain these performance differences. The model is also able to explain differences in performance within the same strategy by identifying the key *Decision Points* during strategy implementation. For example a barrier for stopping the fire can take a variety of forms. The model replicates this variety of forms by allowing blocks of behaviour (microstrategies) compete based on their perceived utility. The impact of the two training programmes over the execution of strategies is further understood using the model. The cognitive model looks for an explanation of this impact using the utility construct: the different stimuli result in different learning outcomes that ultimately produce different rule weights. In a similar way the consolidation of strategies can be further understood. This same approach is used to offer an explanation of how the different training programmes facilitate or hinder the ability of participants to cope with environmental change during the testing phase. Because the model shows the sequence of cognitive, perceptual and motor actions at a fine-grained level it is possible to assess how a change in the environment impacts the execution of strategies. The cognitive model can also shed light on strategy change by describing the process in which the model stops rewarding the use of certain strategies in favour of others. An intrinsic product of developing this cognitive model was the evaluation of different ways of perceiving feedback from the environment. These different ways of being sensible to the feedback of the simulation generate different ways of adapting to the situation. Because the model is able to generate similar behaviours to participants (chapter 6) it is plausible that the model is being sensible to the same kind of feedback as individuals.

### 4.1.2 Scope of the model

The modelling effort is focused on what responses participants emit (their commands) and how long it takes them to emit these responses. There are four areas of interest. The first area of interest is strategy use and implementation and this is modelled using a construct called microstrategies (section 2.5.2). In this case the pattern of command use is the right level of modelling. It is important to say that the lack of other data, such as eye movement patterns, makes the modelling of the timing of commands quite hard. For this reason participants who showed structure in their behaviour only are taken into account when data fitting while very low performers are not considered. Behaviour such as excessive waiting or seemingly random or purposeless execution of commands is not modelled (section 3.4.2 gives more reasons for why a subgroup of participants were selected). The second area of interest is the effect of training programmes on strategy use. Strategy variation is modelled as a combination of preferences for a strategy and the processing of the final feedback in a trial (section 4.3.2.1). This means that both an internal (utility) and an external (FireChief performance) method are used for evaluating strategies. The third area of interest refers to cognitive inflexibility during testing. After the training period the model exhibits inflexibility at two levels: strategy use and strategy implementation, and both kinds of inflexibility can be traced to variations in key production utility values (section 6.1.2.3) produced by the different training programmes. The last area of interest is more specific in nature and refers to the creation of a barrier of Control Fires. The higher level of structure observed in some protocols allows for a more precise modelling of the behaviour related to the creation of the barrier and for this reason stronger predictions are offered in this respect.

As pointed out by Lewis et al. (2004) there is no explicit methodology for how to model task-level behaviour using cognitive architectures such as ACT-R, but although there is no straightforward way of creating models there are a few methodological principles for guiding the development process (Cooper et al., 1996). For example the *Minimal Control* principle (Taatgen, 2005) advocates reducing the number of control states in a strategy. As pointed out by Taatgen a large number of control states increases the risk of brittle behaviour. The risk of brittle behaviour was ubiquitous during the development of the FireChief model. The high level of dynamics present in the FireChief task creates a rich set of combinations of buffer states and environmental situations that increase the risk of arriving at a state the model does not have the knowledge to deal with. The way of implementing a strategy following the minimal control principle is by combining top-down and bottom-up control. Top-down control is used when bottom-up control is unable to discriminate among options.

## 4.2 The design of the model

The core of the model is comprised by the basic workflow (figure 4.1) and the microstrategies described in sections 4.3.1 to 4.3.3. Nevertheless there is a set of features critical for the operation of the model. (1) Strategic governance, as the model must choose a strategy which exerts a weak amount of control over action selection. Section 4.2.3.2 describes how strategies are evaluated. (2) Display quick and recurrent choice behaviour given that in FireChief there are multiple decisions that must be made in a very short timeframe in similar situations (sections 4.2.3 to 4.2.5). (3) Enable the acquisition of interactive skills, as the kind of behaviour required by FireChief can be described as interactive in terms of Fu & Anderson (2008): "learning action sequences in situations that depend critically on the utilization of external cues" (p. 4). See section 4.2. (4) It must have the capabilities described by Larkin (1989) for display-based problem solving. The basic idea is that each action is directly cued by the visible state of the simulation. This last feature is comprised by six elements:

1. Very little must be held in internal WM therefore there is an exploitation of external objects (section 4.2.2).
2. Resilience against errors where WM losses are not important (the content of the imaginal buffer can be correctly updated as it can be encoded from the display). This feature corresponds to the "distraction correction" mechanism (subsections 12 and 15 in section 4.3.1).
3. Little cost for interruption, the only cost is the loss of the internal information related to the current intention (subsections 12 and 15 in section 4.3.1).
4. Little strategic knowledge, where the use of one sequence of actions over another is seldom favoured. The consequence is that the exact sequence depends on two factors: where attention is focused and when working around difficulties is required. This is related to the weak amount of control described before.
5. Adapt behaviour around difficulties. In the FireChief model difficulties are sorted out by adaptation in how strategies are executed and even by changing strategies.
6. Learning is required. The model fires productions to get useful information and determine actions.

The seventh and last critical feature of the model is that it follows the *one-model approach* where the same set of rules is used for all experimental groups, in this way an account of cognitive inflexibility and other phenomena can be provided based on differences in training programmes.

### 4.2.1 Knowledge representation (declarative)

An important design decision implemented in the model is that there is no declarative encoding of relations between actions and outcomes, that is, declarative memory encoding was not pursued. For example, a chunk that explicitly states that a semicircular barrier represents a good option for creating a barrier does not exist. The rationale is that the rate at which decisions are being made renders the effortful maintenance of the consequences of decisions ineffective and also that the highly dynamic environment plus the considerable combination of task configurations and actions makes difficult the definition of a comprehensive set of predefined values for chunks. Another reason is the requirement for the model to be adaptive: reinforcement learning, the basis of ACT-R subsymbolic processing, is more sensitive to error feedback than declarative learning, which is just the kind of adaptiveness required by a model interacting with a dynamic task (Fu & Anderson, 2008). A consequence of leveraging the ACT-R utility learning mechanism for controlling the model's behaviour is that learning tends to be slower in comparison with declarative learning as the utility of productions gradually accumulates the experience as opposed to creating new chunks. The knowledge acquired by means of reinforcement learning is implicit and a result is that the model becomes more sensitive to environmental feedback (Fu & Anderson, 2008). This feedback is mediated by the configuration of the different training programmes in the context of this research.

Nevertheless a set of important declarative chunks is used by the model to create an internal representation of the task. The chunks presented in this section are subjected to the activation mechanism described in section 2.4.1.2. There are chunks for representing goals, strategies, intentions, units, fires, landscape elements, dams, and wind conditions among others. Of all these chunks the first three are the most important and are therefore described in more detail in sections 4.2.1.1-3.

#### 4.2.1.1 Goal chunk

This chunk is stored in the goal buffer and serves to keep track of the current goal by updating the current state of high level intentions. This goal chunk works in combination with the intention chunk (stored in the imaginal buffer). The model progresses through the task by making updates to the slots (i.e. elements) in the goal chunk. The goal chunk has a slot called *intention* and another called *step,* intentions are mapped to microstrategies and each intention has several. For example, imagine that at certain point in time the value in the *intention* slot is *extinguish-fire* and the value in the *step* slot is *find-cell-on-fire*. A rule can now be triggered to update the value of the step slot to *identify-fire-strength*.

#### 4.2.1.2 Strategy specification chunk

By defining the current strategy a strategy specification chunk exerts a high level of control over the model. A particular strategy will impose a pattern of intentions and a way of carrying out these intentions on model behaviour. Because the analysis of participant protocols found four main strategies there are four strategy specification chunks than can produce the whole range of strategies presented in figure 4.6. A strategy specification defines whether the model will use a mixture of DW and CF commands, whether or not a barrier will be created, and which way of attacking the fire is preferred. During a trial the model retrieves a strategy specification chunk several times to help it decide upon its next intention. In ACT-R there is a cost associated with the retrieval of declarative elements, but because this chunk is retrieved frequently the time required for its retrieval tends to get smaller as the task progresses.

#### 4.2.1.3 Intention chunk

The intention chunk is used for tracking the current intention and is stored in the imaginal buffer. This chunk stores a representation of the current situation of the fire fighting process; it holds information about the type of intention, the state of the barrier (if any), the method for attacking the fire and the status of the different units. Units can have one of the following statuses: moving, idle, attacking the fire, issuing a CF command, issuing a DW command, refilling, etc. Other models (such as Lebiere et al., 2001 and Taatgen, 2005) also make use of this module for maintaining information about the position of and actions to be performed by aircraft. In Veksler, Gray & Schoelles (2007) the imaginal buffer is used for storing values that were used during the Table Decision task for making comparisons. Intentions are closely related to the use of microstrategies: different patterns of values in the intention chunk enable the execution of different microstrategies.

### 4.2.2 Searching

In order to explain how a search is performed it is important to briefly describe the interface to FireChief. A LISP (Steel Bank Common LISP) version of FireChief was designed and implemented following the original specification provided in the FireChief manual (Omodei &

Wearing, 1992). Figure 2.6 in section 2.1.1.5 shows the display generated by the LISP simulation. This version is able to control, among other things, landscape behaviour, the development of fire, the execution of commands, and the calculation of performance. After the first view of the landscape is created it is made available to the ACT-R visual module and the model is run. From this moment the display will change according to the FireChief inner mechanics and the actions generated by the model, and the model will perceive these changes using the visual module: the model can switch its attention by firing rules and can recognize objects contained in the experimental display.

A landscape description is represented in LISP as a matrix of strings where different strings refer to different landscape types, units and fires. Considering the training phase, there is a particular matrix for each of the sixteen variable training scenarios and one for the constant training scenario. For the testing stage, when a change is introduced into the environment, the same setup as in the Constant Training condition is used and only wind conditions or appliance efficiency are modified. Figure 4.2 shows one of these matrices of strings.

```
(defvar *terrain-description-CT*
    '(
  "R" "R" "R" "L" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "P" "L" "R" "R" "R" "P" "R" "R" "L" "P" "L"
   "R" "R" "R" "R" "L" "P" "R" "R" "R" "R" "R" "R" "P" "R" "R" "R" "R" "R" "R" "L" "R" "R" "R" "R" "L"
  "R" "R" "P" "R" "R" "R" "R" "R" "R" "R" "P" "R" "R" "R" "R" "R" "R" "R" "R" "R" "P" "R" "R" "R" "R"
   "R" "R" "L" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R"
  "L" "R" "R" "R" "P" "R" "R" "R" "L" "P" "R" "C" "R" "R" "R" "L" "R" "L" "L" "L" "R" "R" "R" "R" "P"
   "R" "R" "R" "R" "D" "R" "H" "R" "R" "R" "R" "R" "R" "R" "P" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R"
  "R" "R" "R" "R" "R" "L" "P" "R" "C" "R" "R" "R" "L" "R" "R" "L" "R" "R" "R" "R" "R" "R" "R" "L" "R"
   "R" "R" "R" "R" "R" "F1" "F2" "F3" "R" "R" "R" "R" "R" "R" "R" "R" "P" "L" "P" "R" "R" "L" "R" "R" "R"
  "R" "R" "R" "R" "R" "F0" "B" "B" "F4" "R" "L" "L" "R" "R" "R" "R" "R" "R" "L" "P" "R" "D" "R" "R" "H"
   "R" "R" "R" "R" "L" "F1" "B" "B" "F0" "F2" "F2" "F3" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "D" "R" "R"
  "R" "R" "R" "R" "R" "R" "F1" "F2" "F0" "B" "B" "B" "F0" "P" "R" "R" "L" "R" "R" "R" "H" "R" "P" "R" "R"
   "R" "R" "R" "L" "T" "R" "R" "R" "F1" "F2" "F2" "F0" "R" "R" "R" "L" "R" "R" "R" "R" "L" "P" "R" "R" "R"
  "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "R" "L" "L" "R" "R" "R" "R" "R" "R" "R" "H" "R" "H" "R"
   "R" "R" "R" "R" "R" "R" "R" "R" "L" "R" "R" "L" "R" "R" "H" "R" "R" "R" "P" "R" "R" "R" "R" "R" "R"
  "R" "R" "R" "R" "T" "R" "R" "R" "R" "P" "R" "R" "D" "R" "P" "R" "R" "R" "R" "R" "R" "H" "P" "R" "R"
   "R" "R" "R" "R" "L" "R" "R" "R" "R" "R" "R" "R" "R" "L" "R" "R" "R" "R" "R" "R" "R" "R" "P" "R" "R"
     ))
```

**Figure 4.2: LISP variable describing the CT scenario**

The visual module identifies the 400 cells in the landscape, the units, fires, wind strength and direction. The result is a table of features called the *Visicon* in which each row represents an element in the experimental window. Every cell in FireChief is represented as two entries in the Visicon: one for the colour of the square and one for the label in the middle of the square. To execute a perceptual action (a Find Location operation) a rule must initiate a search in the experimental window. This search is executed using the “where” system of the Visual Module. The “where” system receives a series of constraints from a production and returns a chunk representing a location. Figure 4.3 shows two productions that specify a series of constraints for the Visicon. In the rule on the left the model is looking for a copter (copters use the colour gray) nearest to the current focus of attention and that has not been recently attended. In the rule on the right the model is looking for a fire (in red) located towards the north-west of the landscape. ACT-R evaluates all the constraints and returns an entry of the *Visicon* that matches the criteria. If no candidate is available the visual-location buffer returns an error. If many elements comply with the constraints one is selected randomly.

```
(P find-unit-for-DW-start-unit-3
       =goal>
       ISA goal
       step select-unit-for-dw
       ?visual>
       state free
       =imaginal>
       ISA intention
       focus2x nil
      ?visual>
      state free
       ==>
       =goal>
       step attend-unit-to-dw
       +visual-location>
       ISA visual-location
       color gray
       :nearest current
       :attended nil
       +retrieval>
       ISA strategy-spec
 )
```

```
(P extinguish-fire-low-fire-start
       =goal>
       ISA goal
       state implement-strategy
       intention extinguish-fire
       step define-focus-DW
       =retrieval>
       ISA strategy-spec
       dw-priority low-fire
       =imaginal>
       ISA intention
       type extinguish-fire
       cell3x nil
       cell4x nil
      ==>
        =goal>
       step define-focus-for-dw
       +visual-location>
       ISA visual-location
       color red
       screen-x lowest
       screen-y lowest
  )
```

**Figure 4.3: Examples of ACT-R rules that search visual elements**

Once an element of the *Visicon* is located in the buffer it is possible to switch attention. The "what" system makes use of the location chunk in the "where" system and executes a shift of attention to that location. When this happens a chunk with the description of the element is placed in the visual buffer. In some cases it is useful to detect whether or not a visual element exists for making a decision; this same approach was used by Byrne (2001). For example when the model needs to determine if a unit finished a command it suffices to detect that a particular colour is present in a given cell by executing a pre-attentive action based solely on the "where" system. Summarizing, in FireChief much of the information required for making a decision is not stored in the goal or imaginal buffers but it is available in the visual field and is accessed via searching and verification rules. Tables 4.2 (b) and (c) in section 4.2.4 show examples of rules used for perceiving the environment.

### 4.2.3 Learning

The FireChief task can be characterized as the acquisition of interactive skills involving recurrent actions that are cued by external information. The FireChief model needs to learn how to stop the fire within a variety of scenarios. For Fu & Anderson (2008) the acquisition of such skill involves the use of internal and external cues. External cues are given in the form of fire and wind behaviour and particularly by displaying the final performance at the end of the trial. Internal cues are mainly in the form of the utility of productions, but there are important declarative elements. As previously mentioned, the behaviour of the model is largely controlled by internal cues, but it is important to say that the context in which the competition of rules occurs is set by the objects continuously perceived by the visual module. The reward scheme implemented in the model is discussed first; the way in which the model uses the final performance is discussed later.

As mentioned in Taatgen et al. (2006) "utility learning is a useful paradigm in tasks where the possible strategies are relatively clear, and where it can be assumed that people already have some sort of procedural representation of these strategies" (p. 49).

The model's ACT-R production rules are the product of the analysis of the participants' protocols (see section 3.2.3) which revealed patterns of behaviour that were encoded as strategies. The firing of rules is the model's engine, the number of rules is high and their utility is continuously modified by rewards from the environment changing their likelihood of being selected again. A practice adopted in this research is to focus the analysis of utility variation on those rules that are instrumental to the model's behaviour: the key rules. The identification of key rules is also used by Janssen & Gray (2012). A key rule is one that enters the conflict set during ACT-R conflict resolution and hence competes in determining the next intention or action of the model. Rules not considered as key are the ones that carry out tasks such as the completion of commands or to carry out perceptual actions that have been specified by key rules. The pattern of utility variation in key rules reveals useful information about what the model is doing while interacting with FireChief. For example, if a key rule increases its utility this means that it contributes to the successful execution of commands; if there are no utility variation then the rule is either receiving stable (both positive and negative) feedback or is not being used.

#### 4.2.3.1 Reward scheme

As Janssen & Gray (2012) put it, the rewarding scheme constitutes the "model's reflection on its performance" (p. 3). Section 2.4.1.5 describes the ACT-R learning mechanisms. From the formula in section 2.4.1.4 is possible to see that the utility of a production is updated by considering the difference between the reward received and the utility of the production prior to the reward being received. This formula also considers an important temporal factor: the reward decreases as more time elapses between the firing of a rule and the moment in which the reward is granted. This temporal difference factor is used to estimate how much a specific rule contributed to the magnitude of the reward. Nevertheless not all productions contribute in the same way to the reward, in many cases some of the most proximal rules to the moment of the reward are not the ones responsible of completing a command, rather the ones that made specific decisions. This kind of problem makes the selection of the appropriate reward scheme a challenging task.

Janssen & Gray (2012) describe different dimensions for characterizing a Reinforcement Learning (RL) problem: the moment, the objective function and the magnitude. Janssen & Gray used the Block World Task described in section 2.3.1.6 to test different combinations of these dimensions and found the strongest effect for the *moment* dimension: giving rewards by-trial vs. by-round basis. The authors propose that different conceptions of rewards lead to different model performance. In a similar fashion different approaches were considered for the FireChief model. Table 4.1 shows different combinations for the relevant dimensions for RL in the context of the FireChief model.

The first reward scheme tested was to give a single reward at the end of a whole trial. This reward depended upon the final performance in that trial. The problem with this scheme was that, because several hundred rules fire during each trial, the temporal difference factor updated the utility value of rules unevenly: those proximal to the end of the trial were able to receive positive feedback whilst the ones at the beginning of the trial were greatly affected by the temporal factor. This approach has been used successfully in other models where participants' interactions require less time (Peebles & Bothell, 2004; Veksler et al., 2007). The second reward scheme tested was based on 'subtasks' defined by the four strategies described in section 4.3. For example, in the Barrier strategy a stage is 'when the barrier is finished', 'when units are refilled (and have issued several DW commands)', or 'when the fire is controlled'. This scheme increased the frequency and therefore the number of rewards and hence the utilities of more productions were affected more evenly. There were two problems with this scheme: the first problem was that, as in the first scheme, various extraneous rules could fire between rewards (for instance, completing a barrier may require around ten CF commands, and for executing them you need at least ten Move commands, and the model can in the meantime use copters to execute DW commands whilst completing the barrier). The second problem was that, as pointed out by Janssen & Gray (2012), there is no explicit reward signal in naturalistic-like settings such as FireChief, where the only one available is the end of a trial. In some trials it was hard to clearly identify subtasks during the execution of strategies, because they are defined by particular changes in the environment, such as the extinction of the fire, and this kind of event does not always occur, for instance, in the Follow strategy it is hard to determine a distinct stage within the continuous execution of DW commands.

| The moment | Objective function | The magnitude | Comments |
|---|---|---|---|
| Once-rewarded | Number of non-destroyed cells | 0-100 (100 means that all the terrain was saved) | Temporal difference factor updated utility unevenly. |
| Subtask-completion-rewarded | The completion of the subtask | 50 success / 0 failure | The FireChief task is not that structured so that specific subtasks can be identified in all cases. Dynamic components and concurrent intentions make harder the credit-assignment process. |
| By-command-completion-rewarded | The successful execution of the command | -For Move: 3 success/ -3 failure<br><br>-For DW: 2 X Fire Intensity success / -2 X Fire intensity failure<br><br>-For CF: 6 success/ -6 failure | Easy to identify moments when reward should be given and the polarity of the reward. Hundreds of rewards are given within a trial. |

**Table 4.1: Different combinations of moments, objective functions and magnitude for reward learning**

In the third approach positive rewards are awarded for successfully completing commands and negative rewards are given for the failure in the execution of commands or the wasting of time. The outcome of a command is observable without considerable delay. For example, when a CF command is issued there is a visual change in the display and the participant can see that the fire cannot spread to the cell in which the command was issued. The magnitude of the reward must be enough to affect the utility value of the rules responsible for taking a course of actions. There are fixed and variable rewards. Executing Move and Control Fire commands generates a fixed amount of reward but the reward of a Drop Water command is a function of the intensity of the fire that is extinguished. The model of Janssen, Gray, & Schoelles (2008) also gives variable rewards based on the number of blocks placed or the speed of placing them in the Block World task (section 2.3.1.6 describes this task). The first two approaches were rejected and the third approach was used.

#### 4.2.3.2 Final reward

The final feedback (amount of terrain saved) represents the most important external cue related to performance in FireChief, this final reward mechanism represents an explicit payoff function in the sense of Janssen & Gray (2012) that enables the model to compare performance for the different strategies. Nevertheless the reward scheme described in the previous section did not consider this piece of information. As a result the model was unable to reflect strategy preference. A FireChief strategy is a complex and long term decision that impacts several actions and cannot be modelled through the selected reward scheme. Rules that select or change strategies have an initial random utility. At the end of the trial strategy selection is rewarded at a global level by considering two factors: the final feedback provided by the simulation (i.e. FireChief performance) and the base performance of that trial (table 4.2). In the VT programme good performance in a harder trial generates a higher reward than the equivalent performance in an easier trial.

To illustrate how final performance is calculated consider two trials: the 13$^{th}$ trial in the VT programme has a base performance of 37.2% whilst the 8$^{th}$ trial has a base performance of 86.8%. Imagine that a final performance of 95% was obtained in both cases using the Barrier strategy. The difference between 100% and the base performance of the trial is considered as the total amount of terrain that can be saved due to the intervention of participants. This means that in much harder 13$^{th}$ trial a participant can save up to 62.8 % of the terrain whilst in the easier 8$^{th}$ trial a participant cannot save more than 13.2%. The proportion of the difference between the performance obtained in the trial (95%) and the base performance (i.e. 37.2% for the 8$^{th}$) considering to the total amount of terrain that can be saved measures the success in that trial. By applying this formula the participant in the 13$^{th}$ trial obtained a score of 92% whilst the participant in the 8$^{th}$ trial a score of 62.1% based on the same final performance. The change in the utility value of the rule that selected the Barrier strategy is modified considering this new score (modulated to keep the utility of productions within a range of -10 and 10) and its current utility value. Manipulating the utility of a production, beyond the standard ACT-R mechanism discussed in the previous section, has also been used in other models such as the work of Schoelles & Gray (2000).

There are other learning mechanisms used by other models that are also used in the FireChief model. As in the case of the Argus Prime Model (Schoelles & Gray, 2000) the FireChief model does not acquire new strategies. Taatgen et al. (2006) stress the importance of offering an explanation of how knowledge is acquired by the production rules; these researchers combine production compilation with instruction to achieve this end. By using the production compilation mechanism, frequent reasoning patterns will generate new rules (Taatgen, 2005). This mechanism produces rules that will link different states of the model and, as a result, “...learning eventually produces rules that move the system from one perceptual input or motor output to the next (as far as they depend on each other), making perceptual and motor actions the main determiner of performance (Taatgen et al., 2005 p. 433). In the case of the FireChief model, strategies are encoded into a set of productions at the beginning of the trial. Nevertheless the opportunity for production compilation is not that frequent in the FireChief task. Production compilation would have been useful for increasing the speed with which actions were executed; however due to the small amount of productions that make retrievals from memory and because most FireChief actions require a perceptual action there are not many opportunities for compiling knowledge. To echo Anderson: “The perceptual and motor actions define the boundaries of what can be composed.” (Anderson et al. 2004; p. 1045). Section 6.2 discusses future lines related to this topic.

### 4.2.4 Executing commands

A CPM-GOMS (Jonh & Gray, 1995) diagram and data extracted from the model is presented here to describe how FireChief commands are executed. A CPM-GOMS diagram illustrates how cognitive, perceptual and motor processes are scheduled and the sequential dependencies among actions. A description of FireChief commands is given in section 2.3.1.5 The model of Veksler, Gray & Schoelles (2007) is able to determine ahead of time which movement will follow and can prepare for it. This preparation can shave seconds off the time required for executing a command. It is harder to implement the preparation of motor commands in the FireChief model in a similar fashion to Veksler, Gray, & Schoelles (2007) because it is harder to know the nature of the next motor action due to the dynamic nature of the environment that makes the whole situation less predictable. It is possible to determine if a command was cancelled (or repeated) by calculating the amount of time elapsing between the sequential issuing of commands for the same unit. If this amount of time is less than the time required for executing a CF (two seconds) or a DW (four seconds) the command is considered cancelled or repeated. Table 4.2 (a) to (f) shows a variety of rules that illustrate different aspects of the model and are referenced throughout this chapter. An example that makes use of all these rules is presented in section 4.4.2 to explain how these rules are combined to produce complex behaviour. Descriptions of how commands are executed presented in the following sections are enriched by referencing these rules.

| **Step 1: determine unit use** | |
|---|---|
| **Rule 1-A:** wait for the truck to move | **Rule 1-B:** switch to the other truck |
| **IF** the goal is to implement a strategy **AND** your intention is to create a barrier with Control Fire commands **AND** you are deciding what to do next after detecting that the current unit is not available **AND** the barrier of Control Fire commands is not completed **AND** the currently attended truck has not issued a Control Fire command yet | **IF** the goal is to implement a strategy **AND** your intention is to create a barrier with Control Fire commands **AND** you are deciding what to do next after detecting that the current unit is not available **AND** the barrier of Control Fire commands is not completed **AND** the last command executed by the other truck is a Move |
| **THEN** wait for the truck to be available | **THEN** evaluate if the other truck has finished its movement |

**Table 4.2 (a): Rules 1-A and 1-B**

| **Step 2: search visual element** |
|---|
| **Rule 2-A**: get visual location |
| **IF** the goal is to implement a strategy **AND** you want to check the status of a truck **AND** you know the coordinates of that truck |
| **THEN** search for a visual-location in that area of the landscape with a light-gray colour |

**Table 4.2 (b): Rule 2-A**

| **Step 3: check if the truck has arrived** | |
|---|---|
| **Rule 3-A**: truck has not finished movement | **Rule 3-B**: truck has finished movement |
| **IF** the goal is to implement a strategy **AND** you checked the status of a truck **AND** the truck has not finished the move | **IF** the goal is to implement a strategy **AND** you checked the status of a truck **AND** the truck has finished the move |
| **THEN** decide what to do next | **THEN** move attention to the Truck |

**Table 4.2 (c): Rules 3-A and 3-B**

| **Step 4: Check status of truck** | | |
|---|---|---|
| **Rule 4-A:** move mouse pointer to truck | **Rule 4-B:** fire detected | **Rule 4-C:** distraction detected |
| **IF** the goal is to harvest the features of a visual object **AND** the visual object was explicitly requested by the model **AND** the truck is ready **AND** in a cell not on fire **AND** the intention is to execute a Control Fire | **IF** the goal is to harvest the features of a visual object **AND** the visual object was explicitly requested by the model **AND** the truck is ready **AND** in a cell on fire **AND** the intention is to execute a Control Fire | **IF** the goal is to harvest the features of a visual object **AND** the visual object was not explicitly requested by the model |
| **THEN** move the mouse cursor to the currently attended cell **AND** continue the intention to execute a Control Fire command | **THEN** find a new location for moving the truck | **THEN** decide what to do next |

**Table 4.2 (d): Rules 4-A, 4-B and 4-C**

| **Step 5: issue a Control Fire** | | |
|---|---|---|
| **Rule 5-A**: execute a Control Fire (successful) | **Rule 5-B**: execute a Control Fire (unsuccessful) | **Rule 5-C:** detect fire |
| **IF** the goal is to issue a Control Fire | **IF** the goal is to issue a Control Fire **AND** the cell is on fire (controlled by simulation) | **IF** the goal is to issue a Control Fire **AND** the cell is on fire (Visicon) |
| **THEN** press the “C” key **AND** register that the truck is issuing a Control Fire | **THEN** press the “C” key **AND** register that the truck is issuing a Control Fire **AND** emit an alarm | **THEN** decide what to do next |

**Table 4.2 (e): Rules 5-A and 5-B**

| **Step 6: Detect alarm** | |
|---|---|
| **Rule 6-A:** Detect alarm | **Rule 6-B:** Ignore alarm |
| **IF** an alarm (a tone) is detected **AND** a truck issued a Control Fire in cell x **AND** the truck in cell x is not executing a Control Fire | **IF** an alarm (a tone) is detected **AND** a truck issued a Control Fire in cell x **AND** the truck in cell x is not executing a Control Fire |
| **THEN** register that an alarm has sound **AND** start the intention of moving the truck to another location | |

**Table 4.2 (f): Rules 6-A and 6-B**

#### 4.2.4.1 Move command (MV)

The approach followed by the FireChief model to execute a Move command is similar to the one of Schoelles & Gray (2000). The first step for executing a Move command is to define a target. The definition of a target depends on the current intention of the model. After the target is defined a unit is selected based on the availability of units (because they can be busy executing a command). Table 4.3 shows the duration of a set of actions involved in the execution of a Move command by the model for different length movements. The movements were completed without the occurrence of alarms or distractions.

| Number | Action | Distance (in cells) | | |
|---|---|---|---|---|
| | | **1** | **4** | **9** |
| 1 | Locate Fire | 0.06 | 0.11 | 0.05 |
| 2 | Store Fire Location | 0.16 | 0.12 | 0.12 |
| 3 | Find Unit | 0.05 | 0.05 | 0.05 |
| 4 | Attend Unit (not in fire) | 0.13 | 0.15 | 0.12 |
| 5 | Move Cursor | 0.38 | 0.46 | 0.50 |
| 6 | Click Unit | 0.37 | 0.35 | 0.32 |
| 7 | Relocate Target | 0.05 | 0.05 | 0.05 |
| 8 | Attend Target | 0.16 | 0.14 | 0.11 |
| 9 | Move Mouse to Target | 0.41 | 0.51 | 0.63 |
| | **Total** | **1.77** | **1.94** | **1.96** |
| 10 | Press Button (finish) | 0.29 | 0.34 | 0.24 |

**Table 4.3: Comparison of different length movements. Time is given in milliseconds.**

Table 4.3 shows that a movement is completed in 10 steps. The critical step is number 9 when the mouse is moved from the initial location to the target location (which takes longer for a distance of 9 cells). Step 5 depends upon the distance between its location prior to the execution of the Move command and the unit selected in step 3. The totals in table 4.3 also show that a difference between a movement of distance 4 and one of distance 9 can be masked by differences in other steps.

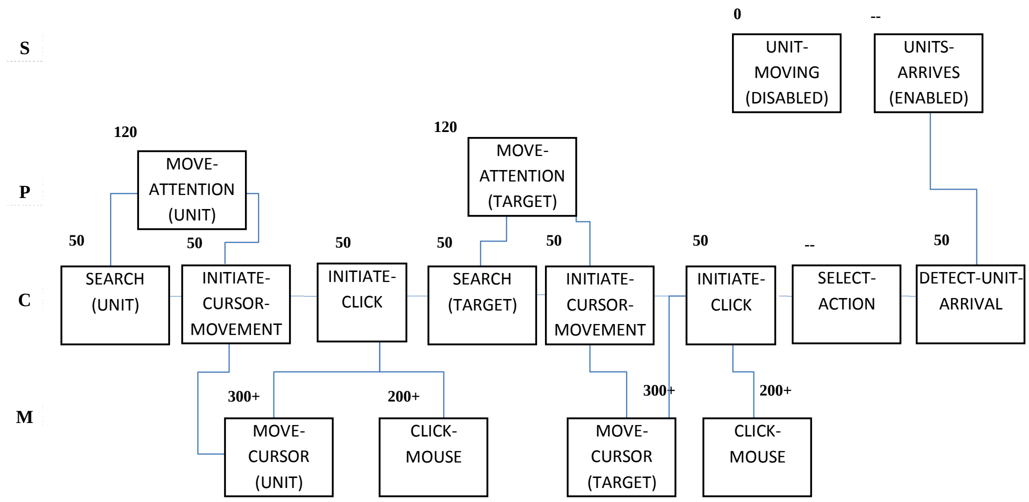



**Figure 4.4: CPM-GOMS diagram of the Move command (S = simulation, P = perception, C = cognition, M = motor)**

Figure 4.4 shows a CPM-GOMS diagram of the execution of the Move command. The relocation of the cursor needs to be completed before initiating the click of the mouse two times. The click of the mouse allows for parallelization: after initiating the first Mouse Click the model starts looking for the target before the click is completed and after initiating the second Mouse Click the model can select an action. This diagram shows how the movement of the mouse is the bottleneck in the execution of the Move command. Table 4.2 (c) in section 4.2.4 shows a couple of rules that can detect weather a truck has arrived or not.

### 4.2.4.2 Control Fire command (CF)

| Number | Action | A (8 cells) | B (2 cells) | C (0 cells) |
|---|---|---|---|---|
| 1 | Confirm that unit is available | 0.1 | 0.1 | 0.1 |
| 2 | Move Attention to unit | 0.149 | 0.13 | 0.111 |
| 3 | Move Cursor | 0.638 | 0.493 | 0.05 |
| | Total | 0.887 | 0.723 | 0.261 |
| 4 | Initiate Key-Press | 0.05 | 0.05 | 0.05 |

**Table 4.4: Comparison of Control Fire commands**

Table 4.4 shows the execution of three CF commands. As it can be seen the critical step is number 3, *Move Cursor*, the length of this movement depends upon its location before executing the CF. In case A the distance that the mouse needs to traverse is 8 cells whilst this distance is only 2 for case B. Case C occurs when the model decides to move the unit, wait for the movement to finish and execute the CF, because the mouse is already in position there is no cost associated to move the mouse (just the 50 ms. associated with the firing of a production vs. the .638 ms. required in case C). There are significant differences in the use of CF between strategies (this is discussed in the next chapter).

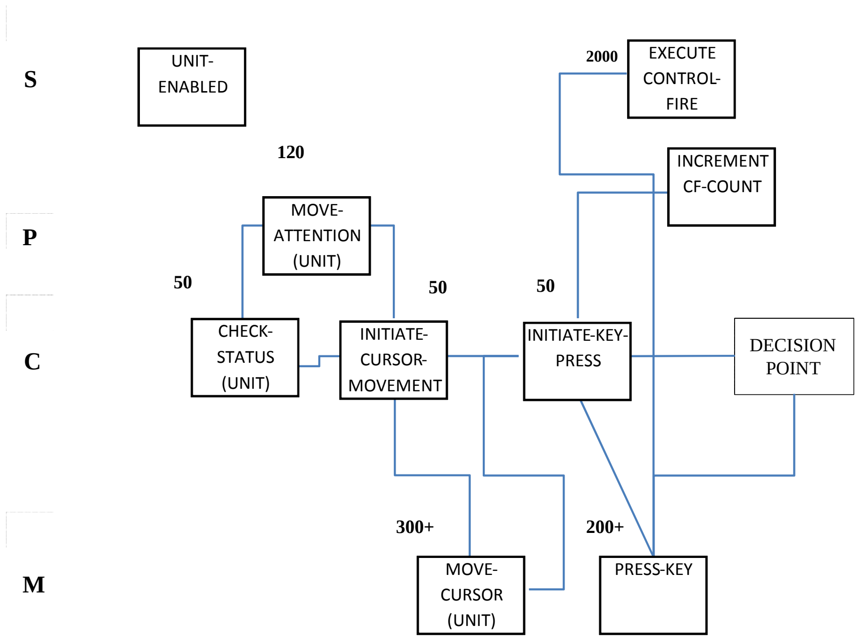



**Figure 4.5: CPM-GOMS diagram of the Control Fire command**

Figure 4.5 shows the CPM-GOMS diagram of the CF command. The motor actions of moving the cursor to the desired cell and pressing the key are in sequence. As in the case of the Move command the relocation of the cursor is the key step for determining the total latency of the command. Tables 4.2 (d) and (e) in section 4.2.4 show rules related to the execution of a CF command if rule 4-A and 5-A are fired one after the other a CF is executed.

#### 4.2.4.3 Drop Water command (DW)

In table 4.5 it is possible to see that the critical action for determining the timing of the DW is the mouse movement. As in the case of a CF the length of the mouse's relocation can be reduced to a minimum. This command shares similar characteristics with the CF command, for this reason the CPM-GOMS diagram is not shown.

| Number | Action | A | B | C |
|---|---|---|---|---|
| 1 | Confirm that unit is available | 0.1 | 0.1 | 0.1 |
| 2 | Move Attention to unit | 0.157 | 0.116 | 0.118 |
| 3 | Move cursor | 0.615 | 0.544 | 0.05 |
| | Total | 0.872 | 0.76 | 0.268 |
| 4 | Press Key (finish) | 0.05 | 0.376 | 0.2 |

**Table 4.5: Comparison of Drop Water commands**

### 4.2.5 Achieving flexibility

In the FireChief task there are four units, three commands and four hundred locations. From a very broad perspective the model's operations are devoted to determining the agent, type and spatial location of the next command and a strategy functions as a mechanism for helping the model to take this decision. The process of making this decision must be flexible enough to consider the dynamic nature of the FireChief task. For this reason the FireChief model does not blindly follow a plan but rather is attentive to the state of the world.

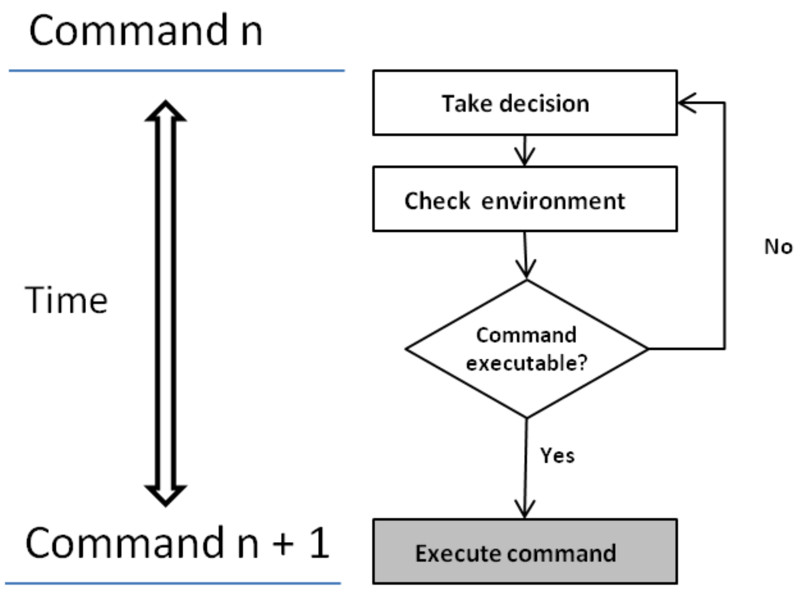



**Figure 4.6: How time between command executions is used**

Figure 4.6 depicts how the model spends its time between command executions undertaking three activities. Taking a decision depends upon the utility of competing productions. Checking the environment to confirm the opportunity for executing commands involves more perceptual actions. Finally, the execution of commands has a significant manual component which requires the biggest share of time. The model is able to adapt to the environment because all its rules were created following the design proposed by figures 4.1 and 4.6. First, the competition of rules is promoted during the stage of taking a decision, therefore the model is continuously faced with the task of selecting a single rule from a pool of several options. Second, there are dozens of rules devoted to sensing the environment, these rules can alter the contents of the imaginal buffer and therefore affect how decisions are made. And third, those rules that increase the probability of successfully executing single commands are favoured.

The status of an ACT-R buffer can be busy, free, requested, un-requested, or error (for example when no elements match the search criteria). Querying a buffer's status (mainly the declarative and visual) allows the model to follow dependency chains. For example, the model needs to wait for an attention switch for checking the intensity of a fire to be completed before deciding how to deal with it. Buffer querying is also the mechanism used in ACT-R for indicating to the model that the element it was looking for is not present.

## 4.3 Modelling strategies

Strategies represent an aggregation level constructed on top of the execution of single commands. Figure 4.7 shows the different levels of aggregation of actions. What is salient is that there is an intermediate layer branded *Microstrategies*. The bottom represents elemental ACT-R actions such as retrieving and modifying chunks or executing manual actions. In this hierarchy the FireChief commands, executed by rules, combine to form microstrategies, which in turn combine to form a full strategy. All strategies are formed from the same global set of elemental activities.

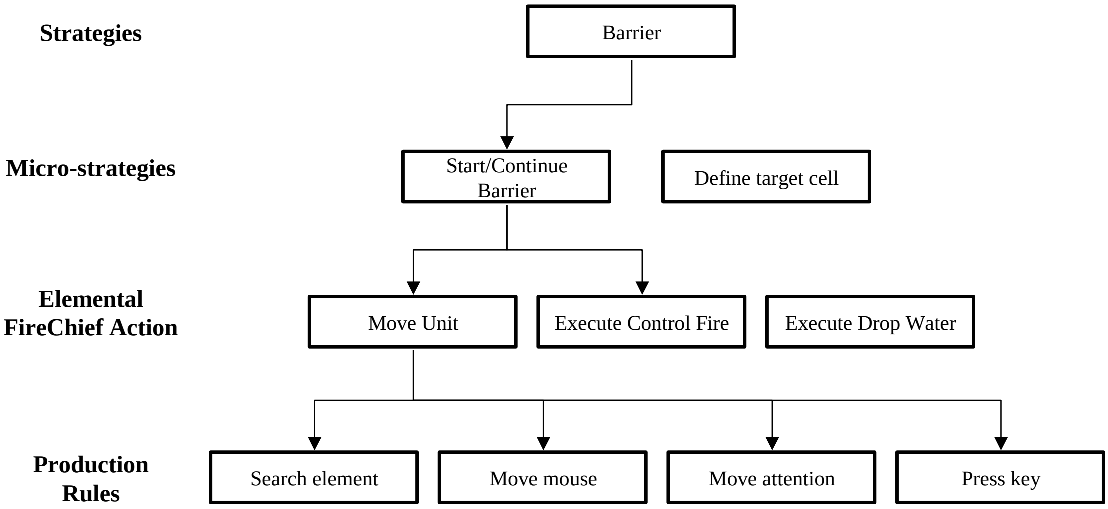



**Figure 4.7: Aggregation levels of actions**

The term microstrategy is taken from the work of Gray & Boehm-Davis (2000) and Schoelles & Gray (2000). According to these authors elemental activities (such as ACT-R commands) can be combined in a finite number of ways to accomplish subtasks called microstrategies. Microstrategies define various ways in which bottom level ACT-R operations can be combined to carry out tasks required to solve the FireChief problem. Section 4.3.1 describes how microstrategies are comprised of *Intentions*. The model can move a unit, select a unit, select the way of using unit, wait for something to happen, search for something, determine the strength of a fire, evaluate the situation, stop the fire by creating a barrier or issue isolated CF commands, drop water over the fire, refill water, and change strategy among others actions.

The general approach for creating the model was to provide it knowledge about microstrategies (in the sense of Gray & Boehm-Davis, 2000) but to allow those microstrategies to compete freely based on their perceived utility. This competition is controlled by the principle of rational analysis embedded in ACT-R. Therefore each strategy is a collection of microstrategies linked by various *Decision Points* (figure 4.1) and coherence in behaviour is provided by a strategy definition represented as declarative chunks stored in the goal and imaginal buffers. As a result the model run is not determined by a strict top down plan but also uses the feedback that the model receives for executing actions producing different patterns of commands (in the temporal and spatial dimension) and, due to the dynamic characteristics of FireChief, these differences impact performance. This approach to model development centred the modelling effort on identifying *Decision Points*, encoding rules for executing actions and assigning rewards.

Considering the high number of visual elements and actions available in FireChief it was possible to create highly complex rules to handle several aspects of the task at once. As a rule of thumb whenever there were several options for defining the productions that comprise a microstrategy the option that offered the simplest solution was used. Simplicity is related to creating rules with the lowest number of chunks and actions involved. This method of creating the model emphasises parsimony throughout the model. Jones, Ritter & Wood (2000) also stress the importance of limiting the number of elements a rule can have (the number of elements to match). By following the principle of limiting rule complexity it was possible to arrive to a set of rules that supports the execution of the various microstrategies whilst preserving cognitive plausibility.

The development of the model underwent many stages. During the first stage it was intended to obtain an ACT-R model able to just interact with the FireChief task and for this to pared down to be the simplest, most minimal model possible. This model represented a starting point based on task goals and environment and prescribed an optimal intervention when dealing with a task while being constrained by the ACT-R architectural features. This initial model had the characteristic of being quite successful mainly because there was a very efficient use of resources: all units were used all the time. Time wasting was effectively absent because the model exploited ACT-R's parallelism and so executed perceptual, motor or cognitive actions at all times. This model followed a strategy definition quite closely, so for example a barrier always had the same form. As expected the data revealed that participants do not use time as efficiently as this initial model nor do they execute the exact same actions over and over again. This initial model served primarily as a basis for constructing an initial set of productions (which were preserved in later versions) for the model and for testing the LISP simulation of FireChief. By looking at how this model interacted with the simulation it was possible to get a feel for the interleaving of cognitive, perceptual and motor operations generated by the FireChief task. This initial model also made evident that it was necessary to increment the number and branching factor of *Decision Points* in order of achieve the desired amount of variability observed in the protocols.

Sections 4.3.1 to 4.3.3 present the analysis of the four main strategies identified in the empirical data. The discussion is structured around the micro strategies that comprise each of these major strategies. The full range of behaviours described in the following sections was incorporated into the cognitive model.

### 4.3.1 The microstrategies of the Barrier strategy

Jones, Ritter & Wood (2000) noted that regular behaviour is easier to model. Because the Barrier strategy produces behaviour that is more regular in comparison with the other strategies it was modelled first. This strategy is described first also because its execution involves a richer repertoire of behaviours and hence a higher number of microstrategies. Figures 4.8 to 4.10 depict the flow diagrams of the four strategies. Blocks are numbered for reference in the descriptions below. A shaded block represents a microstrategy unique to a strategy whilst a non-shaded block refers to functionality shared by many strategies. The flow diagrams do not show all possible alternate flows; rather it is assumed that each block successfully completes its intended actions. The conjunction of the basic cycle (section 4.1) and these microstrategies comprises the essence of the cognitive model.

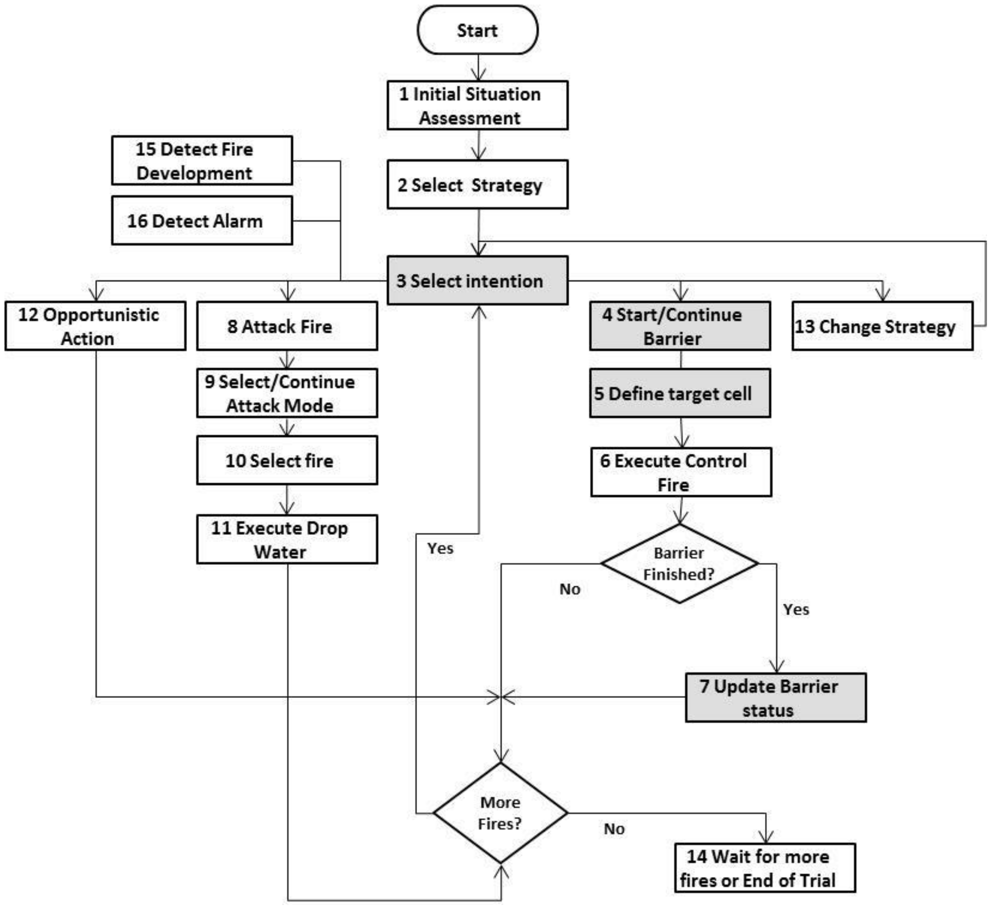



**Figure 4.8: Flow diagram of the Barrier strategy**

## 1 Situation Assessment

Because of the absence of verbal protocols and the fact that only overt commands were registered in the participant protocol logs, it is hard to determine which perceptual and cognitive processes were carried out by participants at the beginning of the trial. Based on the research described in chapter 3 it is expected that participants focus their attention on important aspects of the simulation. Taking as a reference the time needed for individuals to execute the first movement different approaches were tested. In one approach unit and fire locations were attended and chunks representing their characteristics were created. This approach was discarded as it exceeded the time shown by participants and situation assessment was limited to evaluating the distribution of fire. The assessment process locates a fire and creates a chunk to represent this location and rehearsal is used to compensate for the fact that while new fires are located the ones found earlier start losing activation (increasing the risk of their activation level dropping below the activation threshold). The identification of wind strength and direction also happens at this stage and a chunk with this information is created. The order of attention to the fires is guided by the direction of the wind.

## 2 Select Strategy

There is one production for each of the four strategies (Barrier, NonBarrier, Follow and Stop) that when fired sets up that strategy in the goal module. The selection of the very first strategy is largely left to the randomness controlled by ACT-R parameters. What makes these rules different is the additional kind of feedback they receive (section 4.2.3.2).

### 3 Select Intention

Each microstrategy is comprised by one or more intentions. Intentions are the basic block for describing behaviour at the cognitive model level; what generates the model's behaviour is the continuous competition of these intentions. This competition is possible because every intention has a rule representing that intention. The selection of an intention is governed by the current strategy, the current situation and the sub-symbolic level of ACT-R. The current situation and strategy serves as a filter for defining a set of possible productions and in order to select one rule from this set a noisy utility comparison is made. There are various intentions such as *move unit, stop fire, create barrier, extinguish fire, protect landscape, refill tank, evaluate strategy,* and *check wind condition*. An intention is executed by first choosing a focus of attention (called the action cell) and then selecting a command. Ultimately an intention comprises a set of ACT-R commands. The cognitive model is able to change its intention as a result of environmental changes or internal requests. The dynamic characteristics of FireChief interacting with the choices of the model continuously create opportunities for new intentions to be realised or inhibited. This competition occurs throughout the whole execution of the strategy, as a result the model is continuously deciding what to do next. The way of indexing units using the intention chunk is error free in the sense that a unit never loses its index. That is, the model can discriminate between units of the same type at all times so there is less opportunity for mixing intentions. To give a feeling for the number of decisions a model takes over a single trial the list of decisions taken during a single run of the model following the Barrier strategy is presented in table 4.6 where the model made 208 decisions that gave shape to the utility values of the rules involved in taking those decisions. Table 4.2 (a) in section 4.2.4 shows a couple of rules that compete for determining the next intention to follow.

### 4 Start/Continue Barrier

Table 4.6 shows that at second 8.7 the model decides to start a barrier. The model must define a starting point for the barrier. The starting point of the barrier depends upon the current location of the fire and wind conditions. If the barrier is not started in the right place the fire can bypass it before it is finished. The initial point of the barrier also takes into consideration that this initial point must be at a sufficient distance from the fire because the fire must not reach the cell in which a CF is being executed (otherwise it will be cancelled). If the FireChief landscape is considered as a Cartesian plane with the origin at the top-left corner, the right place to start the barrier is a combination of a *y* distance wide enough to span the fire and an *x* distance long enough to leave enough time for creating the barrier. Various cognitive and perceptual actions are required for defining the starting point of the barrier. After this location is identified the model will try to move a truck to it to execute a CF command. If the barrier has been started previously the model can extend it by taking as a reference point the last section of the barrier. Later in the trial the model can decide to start a new barrier in another section of the landscape.

| # | C | T | Choice/Action | # | C | T | Choice/Action | # | C | T | Choice/Action | # | C | T | Choice/Action |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 27 | 4.3 | Select Strategy Barrier | 53 | 406 | 48.7 | change unit | 105 | 767 | 95.7 | issue DW copter 2 | 157 | 1365 | 164.7 | move copter 2 |
| 2 | 29 | 4.5 | Select start attacking the fire | 54 | 411 | 49.5 | issue CF truck 1 | 106 | 775 | 96.4 | wait copter 2 | 158 | 1367 | 165.1 | wait copter 2 |
| 3 | 39 | 7.1 | Move copter 1 | 55 | 422 | *50.5* | issue CF truck 2 | 107 | 858 | 101.7 | move copter 2 | 159 | 1369 | 165.2 | swith to copter 1 |
| 4 | 42 | 7.6 | Wait copter 1 | 56 | 423 | 50.6 | use truck 1 | 108 | 862 | 102.2 | wait copter 2 | 160 | 1372 | 165.4 | check refill copter 1 |
| 5 | 58 | 8.5 | Issue DW copter 1 | 57 | 426 | 51.0 | wait for truck 1 | 109 | 874 | 102.8 | switch to copter 1 | 161 | 1384 | 168.0 | move copter 1 |
| 6 | 60 | 8.8 | Start barrier | 58 | 429 | 51.1 | attack the fire with copter | 110 | 877 | 103.0 | use trucks for attacking the fire | 162 | 1388 | 168.5 | switch to copter 2 |
| 7 | 75 | 11.4 | Move truck 1 | 59 | 443 | 53.8 | move copter 2 | 111 | 887 | 105.0 | move truck 1 | 163 | 1397 | 169.8 | issue DW copter 2 |
| 8 | 76 | 11.7 | Change unit | 60 | 458 | 56.8 | move copter 1 | 112 | 893 | 105.6 | check refill copter 1 | 164 | 1403 | 171.2 | issue DW copter 1 |
| 9 | 91 | 14.3 | Move truck 2 | 61 | 462 | 57.3 | wait copter 1 | 113 | 905 | 108.2 | move copter 1 | 165 | 1412 | 172.0 | switch to copter 2 |
| 10 | 92 | 14.6 | Change unit | 62 | 464 | 57.4 | switch to copter 2 | 114 | 907 | 108.6 | wait copter 1 | 166 | 1468 | 176.8 | move copter 2 |
| 11 | 96 | 14.8 | Wait unit 1 | 63 | 472 | 57.9 | select advanced fire | 115 | 932 | 110.0 | issue DW copter 1 | 167 | 1472 | 177.3 | wait copter 2 |
| 12 | 100 | 15.0 | Use copter 2 | 64 | 482 | 59.9 | move copter 2 | 116 | 938 | 111.1 | issue DW copter 2 | 168 | 1479 | 177.7 | switch to copter 1 |
| 13 | 111 | 17.5 | Move copter 2 | 65 | 483 | 60.1 | ignore fire expansion | 117 | 942 | 111.7 | switch to copter 1 | 169 | 1491 | 178.3 | switch to copter 2 |
| 14 | 115 | 18.0 | wait copter 2 | 66 | 489 | 60.4 | check copter 1 | 118 | 1004 | 116.5 | move copter 1 | 170 | 1497 | 178.7 | issue DW copter 2 |
| 15 | 136 | 19.2 | issue dw copter 2 | 67 | 496 | 61.4 | issue DW copter 1 (too strong) | 119 | 1015 | 117.3 | switch to copter 2 | 171 | 1513 | 181.5 | move copter 1 |
| 16 | 137 | 19.2 | change unit | 68 | 502 | 62.5 | issue DW copter 2 | 120 | 1030 | 120.1 | move copter 2 | 172 | 1518 | 182.1 | wait copter 1 |
| 17 | 149 | 21.2 | move copter 1 | 69 | 503 | 62.6 | switch to copter 1 | 121 | 1032 | 120.5 | wait copter 2 | 173 | 1520 | 182.2 | switch to copter 2 |
| 18 | 153 | 21.6 | wait copter 1 | 70 | 508 | 62.8 | wait for copter 1 | 122 | 1056 | 121.9 | issue DW copter 2 | 174 | 1545 | 185.3 | move copter 2 |
| 19 | 159 | 22.1 | issue DW copter 1 | 71 | 530 | 65.3 | move copter 1 | 123 | 1057 | 122.1 | switch to copter 1 | 175 | 1549 | 185.8 | wait copter 2 |
| 20 | 163 | 22.5 | continue barrier | 72 | 532 | 65.7 | switch to copter 2 | 124 | 1062 | 123.1 | issue DW copter 1 | 176 | 1551 | 185.9 | switch to copter 1 |
| 21 | 168 | 23.4 | Issue CF truck 1 | 73 | 564 | 68.8 | move copter 2 | 125 | 1066 | 123.5 | switch to copter 2 | 177 | 1560 | 187.1 | issue DW copter 1 |
| 22 | 169 | 23.4 | change truck | 74 | 574 | 69.7 | issue DW copter 2 | 126 | 1128 | 128.1 | move copter 2 | 178 | 1566 | 188.4 | issue DW copter 2 |
| 23 | 175 | 23.8 | wait truck 2 | 75 | 576 | 70.0 | continue barrier | 127 | 1132 | 128.3 | wait copter 2 | 179 | 1567 | 188.8 | switch to copter 1 |
| 24 | 180 | 24.6 | issue CF truck 2 | 76 | 593 | 72.3 | move truck 1 | 128 | 1147 | 130.9 | move copter 2 | 180 | 1570 | 188.9 | wait copter 1 |
| 25 | 181 | 24.6 | strick to truck 2 | 77 | 594 | 72.6 | strick to truck 1 | 129 | 1151 | 131.4 | wait copter 2 | 181 | 1632 | 193.5 | move copter 1 |
| 26 | 184 | 25.1 | wait truck 2 | 78 | 599 | 72.9 | issue CF truck 1 | 130 | 1153 | 131.5 | switch to copter 1 | 182 | 1634 | 193.9 | wait copter 1 |
| 27 | 237 | 28.7 | move truck 2 | 79 | 601 | 73.0 | continue attacking fire | 131 | 1168 | 134.0 | move copter 1 | 183 | 1644 | 194.5 | issue DW copter 1 |
| 28 | 238 | 29.1 | stick to truck 2 | 80 | 607 | 74.1 | issue DW copter 1 | 132 | 1172 | 134.5 | wait copter 1 | 184 | 1645 | 194.7 | switch to copter 2 |
| 29 | 242 | 29.3 | wait truck 2 | 81 | 608 | 74.2 | switch to copter 2 | 133 | 1179 | 134.9 | switch to copter 2 | 185 | 1660 | 197.2 | move copter 2 |
| 30 | 247 | 29.6 | issue CF truck 2 | 82 | 621 | 76.3 | move copter 2 | 134 | 1188 | 135.9 | issue DW copter 2 | 186 | 1662 | 197.6 | wait copter 2 |
| 31 | 249 | 29.7 | wait truck 1 | 83 | 625 | 76.8 | wait for copter 2 | 135 | 1194 | 137.1 | issue DW copter 1 | 187 | 1692 | 199.3 | issue DW copter 2 |
| 32 | 266 | 32.2 | move truck 1 | 84 | 631 | 77.2 | issue DW copter 2 | 136 | 1197 | 137.6 | use trucks for attacking the fire | 188 | 1695 | 199.6 | refill copter 2 |
| 33 | 267 | 32.6 | stick to truck 1 | 85 | 634 | 77.5 | use trucks for attacking the fire | 137 | 1208 | 138.9 | issue DW truck 1 | 189 | 1697 | 199.7 | use trucks for attacking the fire |
| 34 | 272 | 32.9 | issue CF truck 1 | 86 | 644 | 79.9 | move truck 2 | 138 | 1221 | 141.4 | move truck 2 | 190 | 1712 | 202.5 | move truck 1 |
| 35 | 274 | 33.0 | use truck 2 | 87 | 650 | 80.6 | switch to copter 2 | 139 | 1224 | 141.8 | wait truck 2 | 191 | 1714 | 202.8 | use copter for attacking the fire |
| 36 | 291 | 35.2 | move truck 2 | 88 | 664 | 81.3 | wait for copter 2 | 140 | 1230 | 142.4 | issue DW truck 2 | 192 | 1734 | 203.8 | refill copter 2 |
| 37 | 292 | 35.6 | wait for truck 2 | 89 | 666 | 81.4 | refill copter 2 | 141 | 1233 | 142.5 | switch truck 1 | 193 | 1740 | 205.7 | move copter 2 to dam |
| 38 | 297 | 35.9 | issue CF truck 2 | 90 | 672 | 83.3 | move copter 2 to dam | 142 | 1249 | 145.2 | move truck 1 | 194 | 1746 | 206.0 | search fires |
| 39 | 299 | 36.0 | use truck 1 | 91 | 673 | 83.4 | switch to copter 1 | 143 | 1257 | 145.9 | refill copter 2 | 195 | 1761 | 206.7 | fire is still burning |
| 40 | 320 | 38.6 | move truck 1 | 92 | 686 | 85.7 | move copter 1 | 144 | 1263 | 147.7 | move copter 2 to dam | 196 | 1767 | 207.9 | issue DW truck 1 |
| 41 | 321 | 39.0 | stick to truck 1 | 93 | 695 | 86.5 | check copter 2 | 145 | 1264 | 147.7 | swith copter 1 | 197 | 1768 | 208.2 | alarm detected insist |
| 42 | 325 | 39.2 | wait truck 1 | 94 | 702 | 87.0 | issue DW copter 1 | 146 | 1275 | 149.9 | move copter 1 | 198 | 1769 | 208.2 | issue DW truck 1 |
| 43 | 338 | 39.9 | issue CF truck 1 | 95 | 704 | 87.2 | use trucks for attacking fire | 147 | 1277 | 150.3 | wait copter 1 | 199 | 1770 | 208.4 | alarm detected insist |
| 44 | 340 | 40.0 | change to truck 2 | 96 | 707 | 87.4 | check copter 2 | 148 | 1286 | 150.9 | issue DW copter 1 | 200 | 1771 | 208.5 | issue DW truck 1 |
| 45 | 357 | 42.5 | move truck 2 | 97 | 711 | 87.6 | change strategy | 149 | 1288 | 151.1 | check refill copter 2 | 201 | 1772 | 208.7 | alarm detected insist |
| 46 | 358 | 42.8 | stick to truck 2 | 98 | 712 | 87.6 | Select strategy Stop | 150 | 1298 | 151.6 | use trucks for attacking the fire | 202 | 1773 | 208.7 | issue DW truck 1 |
| 47 | 362 | 43.0 | wait for truck 2 | 99 | 736 | 90.5 | move copter 2 | 151 | 1305 | 152.8 | issue DW truck 1 | 203 | 1775 | 209.0 | move truck to safe place |
| 48 | 367 | 43.4 | issue CF truck 2 | 100 | 739 | 91.0 | use truck for attacking the fire | 152 | 1318 | 155.4 | move truck 2 | 204 | 1776 | 209.0 | move truck 1 |
| 49 | 368 | 43.4 | switch to truck 1 | 101 | 746 | 92.1 | issue DW truck 2 | 153 | 1329 | 157.7 | move copter 1 to dam | 205 | 1786 | 210.0 | switch to copter 1 |
| 50 | 385 | 45.8 | move truck 1 | 102 | 747 | 92.1 | change to copters | 154 | 1330 | 157.7 | use trucks for attacking the fire | 206 | 1787 | 210.1 | move copter 1 |
| 51 | 386 | 46.1 | switch to truck 2 | 103 | 754 | 92.5 | refill copter 1 | 155 | 1337 | 159.2 | issue DW truck 2 | 207 | 1790 | 210.9 | issue DW copter 1 |
| 52 | 405 | 48.4 | move truck 2 | 104 | 760 | 94.5 | move copter 1 to dam | 156 | 1350 | 161.8 | move truck 1 | 208 | 1793 | 211.1 | Detect alarm end of trial |

**Table 4.6: Decisions taken by the model during a run of 260 seconds**

## 5 Define Target Cell

The data show that the form of a barrier has an important impact on performance. The form of the barrier is a result of the competition for identifying the next cell of the barrier. The barrier can be deployed in a top-down controlled way with a semicircular or linear shape or in a more reactive bottom-up way that depends on the configuration of the fire that is being stopped by the barrier. These two ways of deploying a barrier are inspired by research investigating these two different kinds of control (Taatgen, 2005, section 3.3.2). These two approaches to the creation of the barrier are competing throughout the model run and it is left to the internal mechanisms of ACT-R to decide which rules to apply. In the end the outcome of both approaches is the spatial identification of the next section of the barrier. In both cases the selection of a target cell follows a process in which the candidate cell is proposed and then various tests (based on perceptual actions) are conducted. The selection of a target for a CF command iterates until a suitable candidate cell is found. A candidate cell may be rejected if the fire has already reached it, if a CF command was already completed in that cell or if another unit is currently executing a command in that cell.

| **Number** | **Action** | **Selection method** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Line-1** | **Reactive-1** | **Semicircle** | **Line-2** | **Reactive-2** | **Line-3** |
| 1 | Locate Target | 0.15 | 0.15 | 0.15 | 0.35 | 0.15 | 0.35 |
| 2 | Store Fire Location | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| 3 | Find Unit | 0.08 | 0.06 | 0.06 | 0.11 | 0.08 | 0.09 |
| 4 | Attend Unit (not in fire) | 0.15 | 0.14 | 0.13 | 0.12 | 0.14 | 0.14 |
| 5 | Move Cursor | 0.05 | 0.57 | 0.45 | 0.44 | 0.47 | 0.05 |
| 6 | Click Unit | 0.20 | 0.40 | 0.34 | 0.36 | 0.38 | 0.44 |
| 7 | Relocate Target | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.34 |
| 8 | Attend Target | 0.11 | 0.12 | 0.13 | 0.12 | 0.11 | 0.05 |
| 9 | Move Mouse to Target | 0.59 | 0.52 | 0.40 | 0.58 | 0.46 | 0.14 |
| | **Total** | 1.44 | 2.05 | 1.75 | 2.19 | 1.88 | 1.66 |
| 10 | Press Button (finish) | x | x | x | x | x | x |

**Table 4.7: Sequence of actions for identifying the target cell for the barrier**

Table 4.7 shows the duration of the different activities associated with moving a truck to a cell in order to create a fire-break that belongs to a barrier for a variety of different methods for constructing the barrier. The *Locate Target* action takes longer for the line-2 method shown in the fourth column because the model found that the other truck was already working there. For the line-3 method shown in the last column the *Locate Target* action also takes .35 seconds; the explanation is that in this case the cursor is already in position and moving it to the target takes only 0.14 seconds. What happened in this case is that a fire started in the target position and the model relocated the unit to a neighbouring cell to attempt a new CF. The table also shows the variability in time of the *Move Cursor* movement. Although the latencies for the *Locate Target* action are quite similar the actions themselves are different. What is manipulated by the model in order to locate targets is the scanning area. This idea is inspired by the work of Schoelles & Gray (2000) with the Argus Prime task (see section 3.4.2). A change in the scanning area is realized by manipulating the search constraints for a visual-location chunk. A search of visual-location chunks from the spatial perspective is achieved using two slots: screen-x and screen-y. These slots can be compared against a range of values or by looking for the greatest or lower element among a subset. Other slots such as the type of element and particularly the colour of elements are also used for conducting searches. Using top-down control the next section of the barrier is selected by following a predefined shape. In the case of a semicircular barrier the model considers both locations at the top of the last barrier cell corresponding to curving to the left or the right. From these two locations the model must choose one; this selection depends on the number of CF commands issued before. In the reactive (bottom-up) control approach the model relies only on perceptual actions to select candidate cells. So it will use the previous section of the barrier as a reference point and then will start looking at the fire to determine the next section of the barrier. In the end the resulting shape of the barrier is a combination of top-down and bottom-up control mediated by the ACT-R conflict resolution mechanism. This way of creating the barrier may be maintained throughout creation of the entire barrier if no problems are found. A problem can occur when a CF command cannot be issued because the fire catches up with the barrier, as when a candidate location is on fire or the fire bypasses the barrier.

### 6 Execute Control Fire

See section 4.2.4.2.

### 7 Update Barrier Status

At some point in time the barrier may be completed and this fact changes the behaviour of the model by increasing the use of DW commands both by copters and trucks. A barrier is considered complete if a certain amount of CF commands have been completed with the intention of creating a barrier and the fire has not bypassed the barrier. In this case the imaginal buffer would acknowledge the termination of the barrier goal.

### 8 Attack Fire

All units have the ability to attack the fire, but copters have three advantages: they can extinguish stronger fires, are faster and are not at risk of being destroyed by the fire. The model can use any type of unit to attack the fire but it has a preference for using copters. If the model detects that the goal of creating a barrier of CFs has been completed it will use mainly copters to attack the fire; the rationale behind this is that the strongest fires will collide with the barrier but a lot of terrain can be saved by using the copters to extinguish as much fire as possible. Although used less frequently, trucks are sent to extinguish the weakest fires.

## 9 Select/Continue Attack Mode

The model must choose a way of attacking the fire; a way of attacking the fire serves as a guide for placing constraints on the search for fires to be attacked with DW commands. There are distinguishable ways of fighting the fire. The first one is to *Attack Weak Fires*. In this attack mode fires of low strength (0 to 1) are selected. In a collection of fires burning together with the wind blowing eastwards weak fires are located at the West. The second is *Attack Strong Fires*. In this attack mode fires with high strength (3 to 5) are selected. In the example given earlier this means that the model will look for fires that are located towards the East. These fires are more threatening because they develop faster. Because copters are able to extinguish stronger fires than trucks, it is good practice to use them more frequently for attacking strong fires. The third mode is *Attack Strongest Fires*. In this attack mode the strongest fires are selected first. Using this approach to attack the fire at the beginning of trial can represent a significant advantage: if the strongest fires are extinguished quickly the spread of fire may be significantly slowed. The risk of this approach is that the fire’s strength often exceeds the capabilities of the units. The selection of an attacking mode is not fixed; the model can switch between modes if required. In the case of the Barrier strategy the use of DW commands usually has two functions: to provide more time for creating the barrier and to attack the remaining fires after the barrier is completed.

## 10 Select Fire

This microstrategy is similar to *Define Target Cell*: the approaches to attacking the fire delimit a scanning area for searching for fires. After a way of attacking the fire is selected the next step is to choose a particular fire to attack. The fire selected serves as the target for a DW command. The search for a new fire is a result of placing constraints on a search over the entries of the Visicon. The result of this search is a single chunk representing the target cell. *Select Fire* is more reactive than *Define Target Cell* because a DW command should be issued over one of the fires that are developing in the landscape.

## 11 Execute Drop Water

See section 4.2.4.3.

## 12 Opportunistic Action

Due to the dynamic nature of FireChief it is possible for a combination of events to create the opportunity to execute an action that could benefit the overall process of fighting the fire that is not part of the current flow of intentions. In these cases an *Opportunistic Action* rule can be fired. For example, the model may detect that the tank of one of the copters is depleted and start using trucks to extinguish the fire or send the copter to a dam to refill. Another example is a response to when a CF command is cancelled. The model is able either to move the truck to a new location or alternatively execute a DW command in the same spot. Sending copters for refill is quite important for all strategies. The model must select a dam in order to move the unit. After the dam has been selected the model performs an action quite similar to a movement, the difference is that the unit start refilling its tank automatically when the movement finishes. Refilling usually requires longer movements because the location of dams in fixed and there are few of them. The model can wait until the tank is refilled and then resume the intention that the unit was executing before initiating the refilling. Because a copter's tank capacity is 4 units of water (a truck's capacity is 10) the need to refill the tank occurs quite frequently. Table 4.8 shows the steps required to refill a tank. Note that the detection of the depleted tank is done by an ACT-R pre-attentive check. As in the case of the Move command the time-critical actions are the mouse movements.

| Number | Action | Duration |
|---|---|---|
| 1 | Detect depleted tank | 0.05 |
| 2 | Move Attention to unit | 0.05 |
| 3 | Move Mouse to unit | 0.78 |
| 4 | Click unit | 0.05 |
| 5 | Locate Dam | 0.35 |
| 6 | Move mouse to dam | 0.67 |
| 7 | Total | 1.95 |

**Table 4.8: Refilling a unit**

An important issue relating to the execution of opportunistic actions is how the model "recovers" the intention that was in mid-execution before the opportunistic action arose. The imaginal buffer is used for storing the state of the intention making it possible to identify the current state of a particular intention and continue its flow of actions. Recovery rules are also in charge of allowing the model to keep running after a distraction occurs (see the buffer stuffing mechanism in block 15 of this section).

### 13 Change Strategy

The model can change strategies during a trial if it determines that the current strategy is not effective. It is possible to assess the effectiveness of a strategy by being attentive to the number of burned cells. Table 4.6 shows that the model changed strategy at second 87.6. At this point in time the barrier has been completed and the model replaces the Barrier strategy with the Stop strategy. From this point onwards the model will not try to use CF commands.

### 14 Wait For More Fires or End Of Trial

If the model detects the absence of fires in the landscape it considers the fire to be under control and waits until the end of the trial to check its final performance. If a new fire appears in the landscape in the meantime the model resumes its current strategy. At the end of the trial an alarm will sound and the final performance score is presented to the model.

# 15 Detect Fire Development

The model is aware of changes related to fire behaviour by means of two ACT-R mechanisms. *Scene change* is used for detecting a significant increment of fire development; if such an increment is detected the model can adapt (e.g. change strategy). Determining whether a scene has changed is achieved by looking at the proportion of modified elements from all the elements in the display at a certain point of time and comparing this number to a threshold. The only phenomenon that can alter the environment in such a way that a scene change can be detected is the spread of fire (considering that there are 800 visual elements). The standard way of checking for a scene change is by querying the visual module, but it was not possible to follow this approach due to the fact that the function that refreshes the contents of the Visicon consumes a lot of computational resources (remember that there are 800 visual elements refreshed every 200 milliseconds). The solution was to avoid calling the *proc-display* function (Bothell, 2007) that regenerates all the visual elements in the Visicon and call a customized function instead. This customized function only updates those elements that actually changed during each 200 milliseconds block. A consequence of this workaround was that the value of the scene-change slot is not updated automatically and therefore it was necessary to implement another function to apply the ACT-R scene-change formula. The ACT-R manual (Bothell, 2007) describes the equation used for calculating scene change.

The second mechanism is *buffer stuffing* where a chunk is stored in the visual buffer without an explicit request from a production rule. For the FireChief model *buffer stuffing* represents a recurrent source of distraction for the visual system but also allows the detection of unforeseen events. While the model is engaged in carrying out intentions many things happening in the environment may attract its attention. For example such as when the model has its attention fixed on a cell and a request to change its attention to other cell is started and, before this switch of attention is executed, a fire starts in the current (or adjacent) cell stuffing the visual buffer. The model makes constant use of the information in its buffers for representing the world and executing intentions; that is the reason why an unrequested chunk placed into the visual buffer prompts the model to evaluate the situation. In some cases it is not possible to recover the intention, as when the distraction occurs just after an attention shift was requested stuffing the visual buffer with an invalid visual element. In other cases, for example when a motor command has been initiated, the motor command is executed while the model attempts to recover from the distraction. These recovery rules are activated when a non-requested chunk is placed in the visual buffer. Table 4.9 shows an example of how a distraction causes a delay. At second 49.771, while the model is checking the status of a truck, the model detects that an unrequested chunk is placed by the visual module; the model continues with the intention of checking the truck status (using the recovery rule that fires at second 49.821, this rule uses the content of the imaginal buffer to identify which specific truck it should wait for) and ends up issuing a CF command at second 50.508.

It was considered to alter the way a scene change is detected by the model to increase accuracy. This new scene-change detection method would have a more selective way of evaluating the amount of change. The model’s sensitivity to the development of the fire is controlled by a single number that represents how many elements in the environment must change, between two updates of the visual stimuli, to allow detection by the model. The problem with this approach is that it does not discriminate the visual elements involved in the scene change (a limitation inherent to ACT-R). That is, the total number of cells that are changed due to fire activity is added to the number of cells that are changed due to actions executed by the model, so it is possible that the model is detecting a significant change as a consequence of events generated by its own actions. Because a version of the FireChief microworld was developed from scratch for this research and it was necessary to construct methods to manage updates to the visual memory in a less memory-demanding way, it is technically possible to discriminate between changes generated by the model and changes generated by the simulation, so this upgrade to the model would not require a lot of effort. With this upgrade the model would be more accurate in detecting significant changes in fire development.

| | |
|---|---|
| 49.503 | AFTER-CF-SWITCH-TO-OTHER-TRUCK |
| 49.553 | CHECK-FOR-UNIT-READY-FOR-CF |
| 49.603 | CHECK-FOR-UNIT-READY-FOR-CF-TEXT |
| 49.653 | CHECK-FOR-UNIT-READY-FOR-CF-END |
| 49.771 | CHECK-FOR-AVAILABLE-UNIT-UNREQUESTED |
| 49.821 | CHANGE-INTENTION-NO-UNIT-READY-FOR-CF-WAIT-ACTIONS-FROM-BARRIER |
| 49.871 | CHECK-FOR-UNIT-READY-FOR-CF |
| 49.921 | CHECK-FOR-UNIT-READY-FOR-CF-TEXT |
| 49.971 | CHECK-FOR-UNIT-READY-FOR-CF-END |
| 50.095 | CHECK-FOR-AVAILABLE-UNIT-OK |
| 50.508 | ISSUE-CONTROL-FIRE-ON-SPOT |

**Table 4.9: Example of actions resulting from recovery rules following a distraction**

### 16 Detect Alarm

Alarms can be triggered at any time and the model is able to detect and process them. Alarm detection is carried out through the aural module. This module is similar to the visual module having one buffer for locating sounds and one buffer for harvesting them in the form of chunks. This module also has a list of features called the *Auricon* (which behaves like the *Visicon)*. Section 2.3.1.5 describes alarms in FireChief. Because the alarms emitted by FireChief are all at the same pitch it is necessary to observe the context of the simulation in order to identify the appropriate response. For example if the model detects that the tank is empty the sensible choice is to refill it at a dam. Alarms do not always generate adaptive responses as they can be ignored. Alarm rules were hard to introduce in the model because they can be fired at any time and their interpretation may require information that is not available anymore such as a visual chunk that is lost after a shift of attention. Table 4.2 (f) in section 4.2.4 shows a couple of rules that compete for detecting an alarm or not.

## 4.3.2 The microstrategies of the NonBarrier strategy

The difference between the Barrier and NonBarrier (figure 4.9) strategies resides in the selection of the target cell before issuing a CF command. The remaining microstrategies needed for executing this strategy have already been explained.

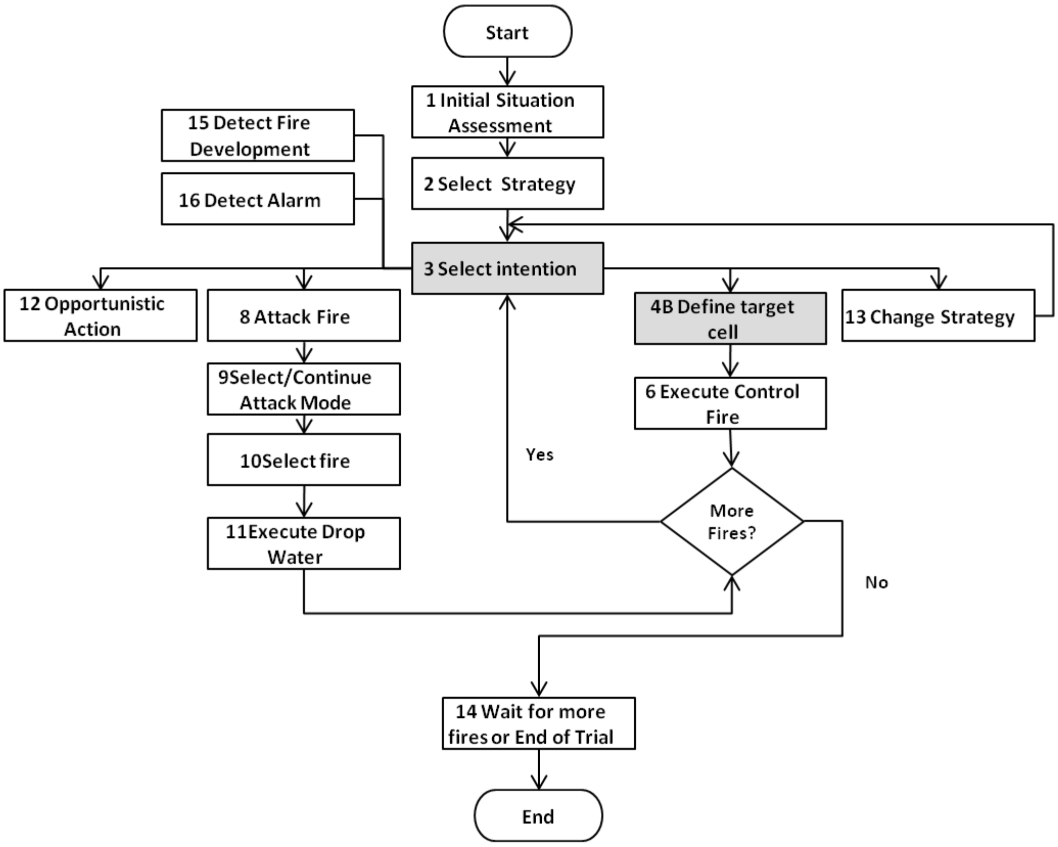



**Figure 4.9: Flow diagram of the NonBarrier strategy**

## 4B Define Target Cell

In the case of the NonBarrier strategy a target cell is identified with the intention of issuing a CF command, but there is no higher goal of creating a barrier. The outcome in this block of activity is a location that will be used for the execution of a CF command. To identify a target cell the model will look for an advanced fire and will select as a candidate target cell a location a certain distance away from this fire. If there are no problems with that location the target cell will be viewed as acceptable and the flow of activity will continue. The *Select Intention* block is also shaded with the intention of highlighting the fact that the strategy definition of the NonBarrier strategy exerts control over the behaviour of other functionality blocks, for example, it exhibits a preference for attacking advanced fires.

## 4.3.3 The microstrategies of the Stop and Follow strategies

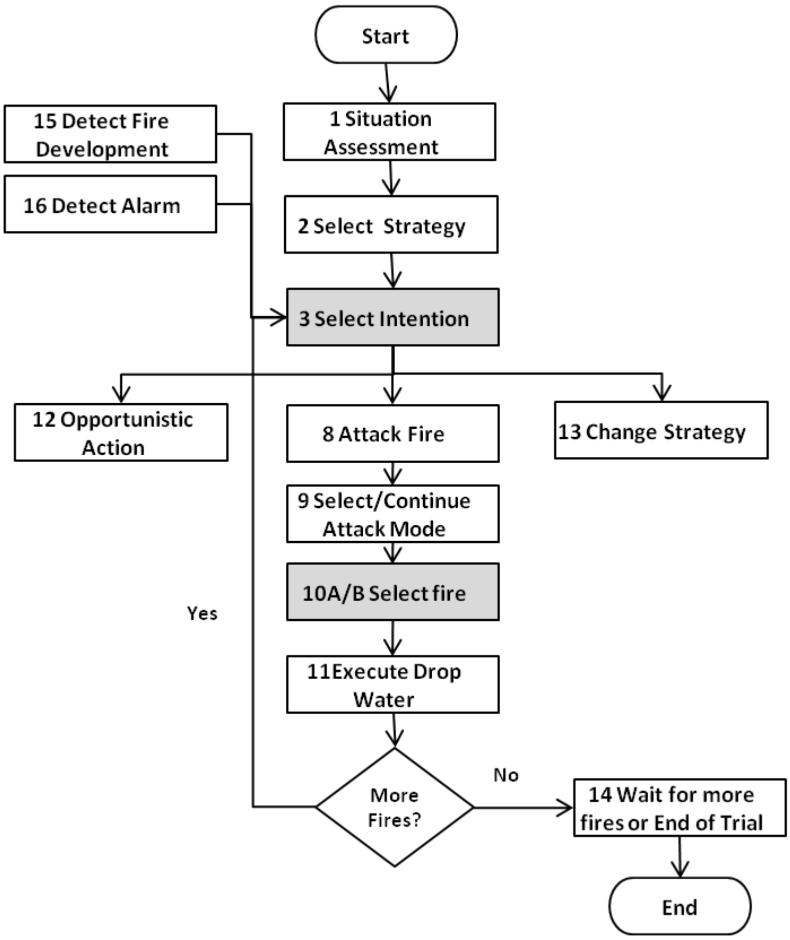



**Figure 4.10: The Stop and Follow strategies**

The Stop and Follow strategies are based on the execution of DW commands. As can be seen in figure 4.10 fewer microstrategies are required for executing these strategies. The block of functionality that determines the difference between Stop and Follow is *Select Fire*.

### 10A/B Select Fire

The selection of which fire to attack is what distinguishes Stop from Follow. This process of selection is more systematic in Stop that in Follow. At the beginning of the trial a model following the Stop strategy will locate and select a few of the strongest fires burning in the landscape and will send the fastest units (copters) to attack them. Remember that the strongest fires are identified during the *Situation Assessment*. The Follow strategy may select a strong fire, but a weak fire is also an option. In Stop, when selecting a new fire to attack, the model uses the previously attacked cell as a reference and tries to extinguish a fire in the immediate vicinity. As fire develops in the landscape it tends to create a round shaped formation where destroyed cells are in the centre and fires burn in the periphery. For this reason execution of the Stop strategy seems like tracking the periphery of the fire with DW commands. If enough DW commands are performed over frontal fires the spread of the fire can be halted. In Follow the previous location of a DW is not considered, only the location of fires. This implies a less systematic way of attacking the fire. Although many fires can be extinguished using the Follow strategy fire development is not stopped.

### 4.3.4 Comparing strategies: a cognitive modelling perspective

The effort-accuracy framework (Payne et al., 1993) suggests that the structure of FireChief determines the quality of the strategies and interacts with the cognitive capabilities of the problem solver to determine the required cognitive effort for each strategy. The average number of productions used by a key microstrategy for each of the four strategies is presented in table 4.10. The strategy with the highest cost is Barrier while Follow has the lowest cost. Besides being more structured, strategies Barrier and Stop also show a higher level of cooperation among units in comparison with NonBarrier and Follow. For instance, during the creation of the barrier the behaviour of one truck usually depends on the other truck's actions and in Stop a determinant of the next action is the current location of the unit and the fire situation. The effectiveness of CF commands for stopping the fire is positively correlated with the degree of proximity among them. When participants (or the model) are creating a barrier not only are the CF locations close to each other but they are also purposely distributed considering the morphology of the fire and the direction of the wind. The creation of a barrier increases the demands of the search for the next target cell.

| Strategy | Microstrategies | Average number of rules |
|---|---|---|
| Barrier | 16 | Define target cell: 12 |
| NonBarrier | 14 | Define target cell: 5 |
| Stop | 12 | Select Fire: 8 |
| Follow | 12 | Select Fire: 4 |

**Table 4.10: Average cost of key microstrategies**

The Barrier strategy on average takes more cognitive steps than the NonBarrier strategy when issuing a CF command, but the manual actions of the NonBarrier strategy take longer. It is also observed that a CF issued in the construction of a barrier takes less time than a CF command issued outside a barrier. The reason behind this phenomenon is that the trucks involved in the creation of the barrier are closer to each other because the previously executed CF command serves as a reference point for the next block of the barrier. Also the barrier is normally constructed by issuing CF commands sequentially. Consequently the trucks creating the barrier tend to be in close proximity to the cell they will next be moved to, reducing the time needed for moving the mouse pointer and hence the overall time needed for executing the CF command. In the case of the NonBarrier strategy CF commands are more dispersed from each other and their execution interleaves more with the execution of DW commands, increasing the distance that the mouse pointer needs to travel between commands and therefore overall. Table 4.11 shows the action times for executing a Move command for six CF commands executed for a barrier and six executed when not constructing a barrier. The model demonstrates more variability in the cognitive and perceptual actions required for selecting the next section of the barrier. The amount of processing depends on the proximity of the fire, the location of units and the wind strength. The model also demonstrates shorter mouse movements and shorter times for movements executed when creating a barrier.

| | Strategy | | | | | | | | | | | | Averages | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Action** | **Non-Barrier** | | | | | | **Barrier** | | | | | | **NBA** | **BA** |
| Locate target | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.15 | 0.15 | 0.15 | 0.35 | 0.15 | 0.35 | 0.20 | 0.22 |
| Store Fire Location | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Find Unit | 0.08 | 0.08 | 0.06 | 0.09 | 0.06 | 0.06 | 0.08 | 0.06 | 0.06 | 0.11 | 0.08 | 0.09 | 0.07 | 0.08 |
| Attend Unit (not in fire) | 0.12 | 0.15 | 0.13 | 0.14 | 0.12 | 0.14 | 0.15 | 0.14 | 0.13 | 0.12 | 0.14 | 0.14 | 0.14 | 0.14 |
| Move Cursor | 0.70 | 0.77 | 0.52 | 0.55 | 0.72 | 0.97 | 0.05 | 0.57 | 0.45 | 0.44 | 0.47 | 0.05 | 0.70 | 0.34 |
| Click Unit | 0.30 | 0.37 | 0.37 | 0.33 | 0.34 | 0.39 | 0.20 | 0.40 | 0.34 | 0.36 | 0.38 | 0.44 | 0.35 | 0.35 |
| Look Target | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.34 | 0.05 | 0.10 |
| Attend Target | 0.11 | 0.14 | 0.12 | 0.14 | 0.11 | 0.16 | 0.11 | 0.12 | 0.13 | 0.12 | 0.11 | 0.05 | 0.13 | 0.11 |
| Move Mouse to Target | 0.81 | 0.71 | 0.49 | 0.72 | 0.71 | 0.53 | 0.59 | 0.52 | 0.40 | 0.58 | 0.46 | 0.14 | 0.66 | 0.45 |
| **Total** | 2.42 | 2.51 | 1.98 | 2.28 | 2.37 | 2.55 | 1.44 | 2.05 | 1.75 | 2.19 | 1.88 | 1.66 | 2.35 | 1.83 |

**Table 4.11: Comparison of Move commands before issuing a CF command for a barrier and when not constructing a barrier. The time is measured in milliseconds.**

Executing a CF command requires more effort (i.e. there are more rules involved) than a DW command. This means that the Stop and Follow strategies should benefit from the effort saved by issuing mostly DW commands. Because a DW takes twice the time to complete compared to a CF, the former offers a bigger time window for using other units. The use of the DW command differs between the Stop and Follow strategies. DW commands issued while following the Stop strategy take less time than DW commands issued while following the Follow strategy. There is also a difference between using trucks and copters: DW commands issued by copters take less time than DW commands issued by trucks (section 6.2.7). The Stop strategy focuses on the strongest fires and therefore copters are preferable; because the proximity of fires targeted by the copter is closer the time relating to moving the mouse is minimised. Only two participants in the CT condition were able to execute Stop with success due to the high wind strength, the model was able to do it as well a few times. In the VT condition Stop is used frequently and if the wind strength is low this strategy generates excellent results. In general terms the experience of participants/model with the Stop strategy differs considerably between training groups. This is important because in the unit efficiency-reduction testing condition using the Stop strategy may be disastrous.

A frequent decision is about what to do next after a unit has been moved with the intention of issuing a CF or a DW command. Just after initiating the last click for issuing the movement command the model must decide whether it will keep its attention on the same unit or whether it will move attention elsewhere. In either case a positive or a negative reward will be awarded depending on the outcome of the command. The decision about what to do next has an important impact on performance, and is reflected in the way units are used. There are two broad ways in which units are used: either sticking to one unit or alternating between them. Taking as an example the creation of the barrier, if the model chooses to alternate between units it may start by moving both trucks in order to execute two CF commands for constructing the barrier. Because these two commands are executed in sequence and the speed of the trucks is low (comparing with copters) both units will be disabled at the same time because they are moving and the model may decide to wait. When the model detects that one of the trucks has arrived at its destination it switches attention to that unit, places the mouse over it and executes a CF command that will finish after two seconds. Because the model is using both trucks, after issuing the first CF command, it then switches attention to the other truck (which by then has probably arrived) and issues the other CF command. If the model waits for a unit to arrive at its destination it will waste a certain amount of time, but the command, CF or DW, will be executed as soon as the unit arrives, increasing the probability of executing the command successfully. If the model switches its attention to other units instead of waiting, a waste of time is avoided if the newly attended unit is available; but the situation may arise in which the first unit arrives at the target location and remains idle for a while before receiving further commands. In such situations the probability of completing a command is reduced (because the fire is constantly developing). As the model interacts with the simulation it must make these kinds of decisions continuously and the outcome of these decisions is reflected in the utility values of its productions. Using many units at a time increases the number of cognitive and perceptual actions, but usually improves performance because waiting times are reduced. Unit use is governed by the competition of productions. To support this, the model maintains an index for each unit, so it is able to distinguish between the two copters and the two trucks. This index is stored in the imaginal buffer. The idea of an index for tracking multiple objects is taken from the FINST model (Pylyshyn, 1989).

### 4.3.5 Patterns of behaviour not covered by the model

The most important feature considered during the development of the model was how to enable it to adapt to the ever-changing FireChief environment. Ceteris paribus, observed variations in strategy are the product of the runtime selection of a particular microstrategy based on rule utility strength resulting from its interactions with the trial scenario. Chapter 3 presents the hierarchy of strategies found during the analysis of the data. These strategies cover the most frequent patterns found in the data and it was a design decision to focus modelling on these patterns and as a result the model captures the most prevalent strategies and intentions observed. Nevertheless patterns not covered by these strategies were found. For instance, in several participants' protocols it can be observed that, after the fire is under control, various commands are issued "randomly" at different locations in the landscape. The model does not exhibit this behaviour because as soon as the fire is under control the main goal is completed. This is an interesting behaviour that is not accounted for in the current model, it may be an emergent phenomenon owing to the uncertainty in the task, where spot fires may appear at any time.

Another behaviour not reproduced by the model was executed by one of the participants on several occasions: a horizontal barrier, completed to the north of the original fire (see the sequence of CF commands, in red, issued in a straight horizontal line in Figure 4.11), affects the fire's behaviour throughout all the test trials of the CT programme (when the wind changes direction). The fire will switch its direction first to the NE at second 80, and this barrier extinguishes some of the fires that start spreading with more strength in that direction. When the wind switches its direction to the N at second 120, more fires collide with this barrier. This barrier is not that effective when the wind changes to the north-west at second 160 and later to the west at second 200; notice that this participant finishes the trial fighting the fire at the north-west of the horizontal barrier. The model never produces a pattern of behaviour similar to the one depicted in figure 4.11, it has the knowledge to do it, but other behaviours were preferred instead. It seems that the participant has the goal of creating a horizontal line of CF commands. It may be the case that this participant failed to discover the Barrier strategy. Why people show these behaviours needs further investigation. Section 6.2 discusses other topics that can be further explored with the model.

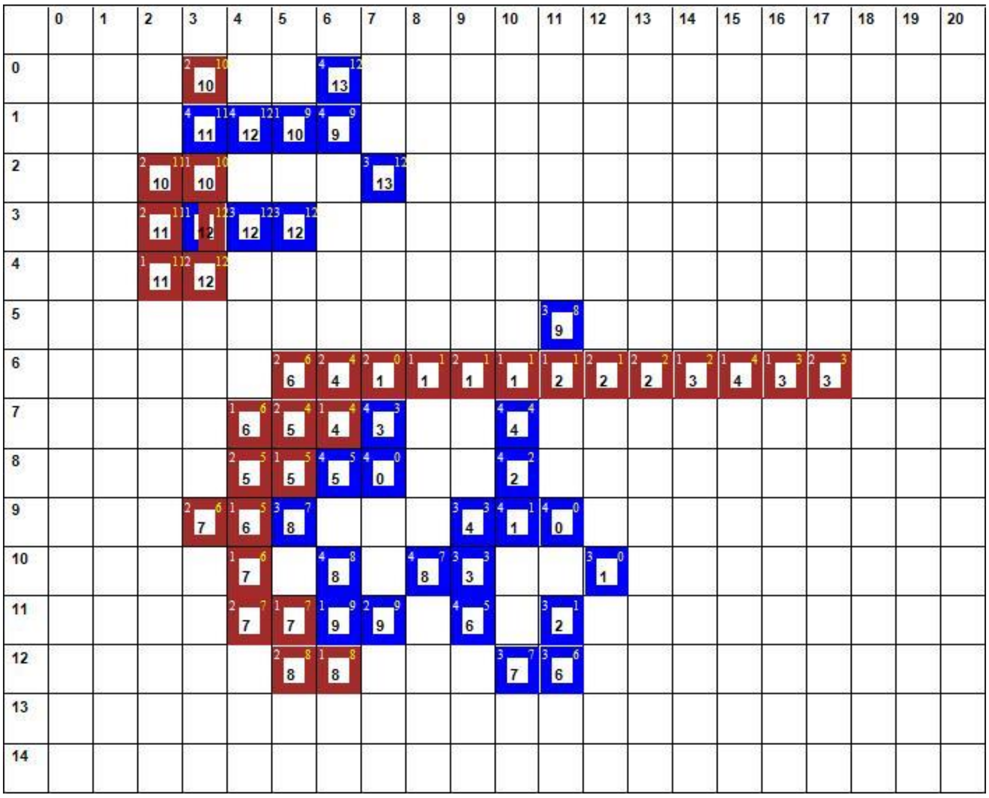


**Figure 4.11: Participant creating a horizontal line of CFs**

## 4.4 Running the model

Each ACT-R model run is meant to be equivalent to a participant's intervention over an entire sequence of training and test trials. To this end the model does not behave deterministically but, by being constrained by the ACT-R architecture, there is meaningful non-determinism (Lebiere et al., 2001). The protocols generated by the model are processed and stored in a database in the same fashion as those generated by participants thus allowing their analysis, including the generation of views (section 4.2.2). Anderson et al. (2004) argue that strong constraints on parameter values are necessary in order to make real predictions about performance. The model therefore uses the values considered as standard by the ACT-R research community. The most relevant parameters include the activation of subsymbolic computations and utility learning, and a value of 3 for the expected gain noise parameter used during the conflict resolution stage. Fitting the model was limited to manipulating the location (and size) of rewards (table 4.1).

### 4.4.1 Model output

The model is run by calling the following LISP function:

*(run-participant <identifier:string> <training-type:symbol> <testing-type:symbol> <initial-trial:int> <final-trial:int> <trial-duration:int> <debugging-mode:t|nil>).*

An example of this function's call is: *(run-participant "M01" 'C 'W 1 24 260 nil).*

The first parameter is the identifier for the data to be generated. The second is the type of training; either 'C for constant training or 'V for variable training. The third is the type of change in the environment during the testing phase; either 'W for a change in the wind or 'E for a change in unit efficiency. The fourth and fifth parameters define the starting and ending trial numbers respectively. The sixth parameter controls the amount of time that the model will be run per trial and the last parameter controls whether the model is to be run for debugging or not. A run of the model generates 3 files:

1. A protocol of commands in the same format as the original FireChief simulation with the information about the timing, location, unit, landscape, performance and sequence of commands.
2. The utility value of productions for each trial. Utility values are recorded when the trial is finished and are used for determining how the model is tuning its behaviour to the characteristics of the task as rewards are awarded.
3. The pattern of strategy use for each trial for keeping track of which strategies are selected for each trial.

### 4.4.2 Putting it all together: a close-up view of the model running

The final cognitive model is comprised by the following features: (1) the same set of rules is used for all experimental groups and experimental conditions (2); a model run starts by explicitly selecting a strategy; (3) the model can change strategies during a trial; (4) the model does not acquire new strategies but the way in which the available strategies are implemented is variable; (5) at the end of the trial strategy selection is rewarded (section 4.2.3.2); (6) the mechanisms that allow a weak amount of control over action selection is described in the following sections 4.3.1 to 4.3.4, strategies are specified using the Strategy Specification chunk (4.2.1.2); (7) the model adheres to the basic cycle which produces quick and recurrent choice behaviour, the Intention chunk supports this feature (section 4.2.1.3); (8) there is a strong reliance on perceptual actions, this means that several actions are cued by the visible state of the simulation (section 4.2.2); (9) the model is able to suffer distractions and recover from them; (10) positive rewards are awarded for successfully completing commands and negative rewards are given for failure in the execution of commands or the wasting of time

As an example of how productions compete to generate complex behaviour, Table 4.2 (a) to (f) presents a detailed description of the rules referenced in Figure 4.12. The context of the example is as follows. The model has chosen the Barrier strategy and decided to start attacking the fire, so the imaginal buffer registers the intention to drop water over the fire. A *Decision Point* arrives when the model decides which particular way of attacking the fire will be followed. Because the model is following the Barrier strategy the possible ways of attacking the fire are constrained. At this point the model chooses to attack the most threatening fires located at the front of the fire (the east). The model locates one of these frontal fires by making a search of the landscape or by retrieving a stored chunk created during the initial assessment of the situation. The next step is to select a unit for attacking the fire: the model selects a copter. When the copter is found a series of motor commands are performed for moving the copter to the desired fire. When the last manual action for the movement is finished the copter initiates its movement towards the fire and the unit is disabled for some time generating a new *Decision Point* to decide between waiting for that unit to finish its movement, attacking another fire with another unit to pursue a different intention. In this example the model decides to start a barrier of Control Fires to the east of the fire. This mixture of dropping water while the barrier is created is well rewarded: by attacking the most advanced fires with copters precious time is gained and that allows the model to finish a barrier. In the example shown below the model has just started a Move command for one of the trucks so the FireChief simulation has disabled that unit. The other truck is ready to receive commands.

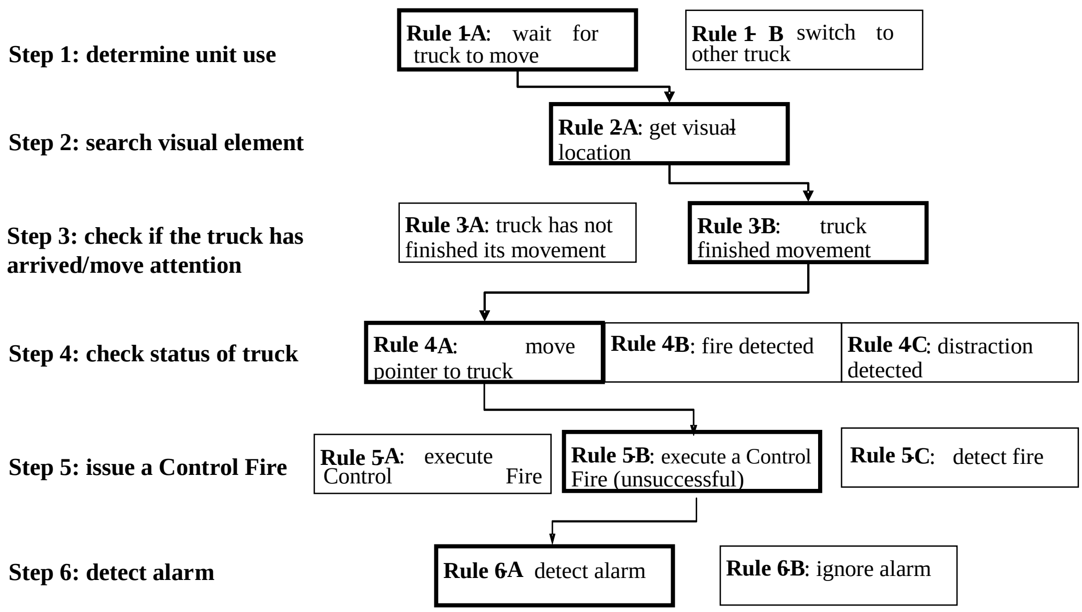



**Figure 4.12: The FireChief cognitive model running six cycles**

Figure 4.12 shows the firing of 6 rules. In step 1 the model must choose between waiting for the truck that has been recently moved and using the other truck. In this step the utility values of productions 1-A and 1-B are compared and the one with the highest utility value is fired. In this example rule 1-A is fired so the model waits for the truck to arrive at its destination. After the model decides to wait it searches for a visual-location that satisfies the specific set of constraints specified by rule 2-A and the result of this search determines step 3. During this wait the model enters a loop between rules 3-A and 1-A: the model decides to wait, senses the environment and finds that the truck has not finished its movement, each time this occurs a negative feedback is received, this feedback reduces the utility of production 1-A. therefore, if the movement takes long enough, rule 1-B would eventually win the competition over rule 1-A. Eventually the truck arrives at its destination and therefore the rule 3-B is fired, the arrival of the truck triggers a positive reward that increases the utility of production 1-A. In this example although the utility of production 1-A is reduced production 1-B is never fired. The model's attention is shifted to the current location of the truck which is now ready to execute a CF command. The visual buffer is now loaded with a chunk representing the content of the cell, that is, the model knows the type of terrain and that there is truck there. In step 4 it is important to verify that the chunk in the visual buffer is not un-requested (see the explanation of rule 4-C in table 4.2, subsection d). If the visual element encoded in the visual buffer is a product of an explicit shift of attention, as in the case of this example, then a mouse movement is initiated towards the attended cell (provided the cell is not on fire; otherwise rule 4-B would fire). In step 5, after the mouse movement is completed, the CF command is initiated by pressing a key. In the normal flow of events the CF command would start after the click of the key is completed and after two seconds the CF command would be completed. In the example here there is a different outcome: just after rule 5-A fires the target cell catches fire preventing the execution of a CF command. The result is that when the click of the key occurs after around 350 milliseconds, instead of starting a CF command the simulation generates an alarm. Because the CF command was not successfully executed a negative reward is given. This negative reward affects the utility of all the productions that fired after the last reward (1A, 2A, 3B, 4A, and 5B) where some of these productions are key (section 4.2.3). At this point the model can detect the alarm and, making use of the contents of the imaginal buffer, can select a course of action. In this example there is competition of productions in steps 1, 5 and 6 (these are decision points). In step 1 the only source of knowledge for the model in making a decision is the previous experience of the model in the same situation. The recorded utility of the winning rule is modified according to the feedback. In this example production 1-A receives negative feedback for waiting and a positive feedback when the truck arrives. Step 3 and step 4 are driven by perceptual actions that are querying the state of the simulation. Step 5 is mainly driven by the simulation (rules 5A and 5B) which controls the execution of commands, but a rule that checks whether the cell caught fire just before executing the command (rule 5C) may be fired and therefore the CF command is not attempted. In step 6 the model compares utility values to decide whether to attend to the alarm or not. As explained in section 4.4.4 not all alarms are processed by the model. Sometimes the tone is detected after the firing of many productions has changed the state of the imaginal buffer so that by the time the alarm is detected the model is unable to identify the source of the alarm.

There are external factors that increase the complexity of pattern matching. For example rules 5-A and 5-B are activated by the same patterns in the ACT-R buffers but also take into account the presence of fire in the cell which is not queried by inspecting the buffers but by inspecting the simulation's state. In both rules the model issues a key-press but the resulting behaviour is different: in rule 5-A a CF command is issued while in 5-B an alarm is emitted. This kind of situation occurs frequently during the running of the model due to the dynamic nature of the FireChief task. If the model is successful in stopping the fire it can wait until the trial ends (the trial lasts 260 seconds) and will receive a number representing its final performance; if this number is high the utility of using the Barrier strategy (the only strategic rule that is affected is the one that selected the Barrier strategy, see section 4.2.3.2) is increased, increasing the chance of selecting the Barrier strategy in the next trial.

## 4.5 Summary

A considerable amount of LISP code is required for displaying and controlling the FireChief task, and for allowing the interaction between ACT-R and FireChief (this was the most difficult part of the model to implement from a technical point of view). As predicted by St. Amant, Freed, & Ritter (2005) the FireChief model re-uses similar sets of rules in many places. A logical structure to the rules was enforced throughout the development of the model. Rules are grouped by functionality and scope (see figure 4.7) making it possible to reuse parts of its code in tasks that present similar characteristics. To keep track of the large set of productions, around 900, is a major difficulty. Some coding standards were followed, the naming of productions rules being the most important. These names are unambiguous and large enough to group (according to functionality) and discriminate rules.

Among other topics this chapter described the design considerations embedded into the model (section 4.2), the micro strategies implemented in the model (section 4.3) and provides detailed examples of the rules used by the model (sections 4.2.4 and 4.4.2). The model continuously interleaves cognitive with perceptual-motor operations, selects different strategies and implement them according to the reward structure of the task. In making a decision about how to use a unit (the most important resource) the model can either follow a plan (follow a less perception-intensive approach) or seek more information from the environment. In the second case the model needs to determine how much information will be gathered before making a decision. As the model interacts with the FireChief task it learns to make these kinds of decision based on the rewards it receives from the environment. The FireChief model is able to deal with a complex, dynamic task by following a coherent set of principles that can be extended to other domains (this is discussed in chapter 7). By providing a loose strategy definition the model is able to implement by itself complex patterns of behaviour which in turn are able to successfully stop the fire while replicating many aspects of the data. The following chapter presents the results obtained by running the model and evaluates how well it captures the interactions observed in the participant data.

# 5 Results

The ultimate objective of this research endeavour is to increase our understanding of strategy use in dynamic tasks: how strategies develop and are selected, improved and changed. Chapter 4 described how the cognitive model works; the objective of this chapter is to show how well the model was able to replicate the empirical data. A particularly relevant section of this chapter describes how the procedural knowledge embedded into the model, governed by a set of utility values, was tuned to the task by the continuous interaction of the model with the simulation and, among other things, how the different training programmes mediate this process. As was mentioned in the previous chapter, knowledge about how to execute actions is provided to the cognitive model, but the exact emergence of strategies is a product of the interaction between the problem solver and the environment. This chapter presents the comparison between the data obtained by running the model and data obtained from participants (presented in chapter 3). The intention of comparing these two sets of data is to demonstrate that the model was able to capture various aspects of participant behaviour. This chapter also presents insights regarding the research questions posed in chapter 2.

This chapter starts by describing how data was generated and how the model's quality of fit (QOF) with the data was determined. In the following section results are presented using the criteria introduced in chapter 3 (section 3.4). This section starts with a description of strategy use in the different experimental conditions and continues with a discussion about command executions, timings (latencies) and adaptivity issues. In section 5.3 results are examined in relation to the topics of interest: strategy use, the existence of cognitive inflexibility, how actions are selected, and how the model copes with the cognitive demands of a complex dynamic microworld. The last section presents a view of a good performer based on what was learnt with the model.

## 5.1 Generating model data

The model generated the same kind of data as participants whilst providing a detailed trace of the operations being executed inside its various modules. By combining the information provided by the different ACT-R modules with the knowledge obtained during the analysis of the FireChief task, explanations related to the use of strategies were constructed.

| | Participants | Model |
|---|---|---|
| **CTW** | 8 | 10 |
| **CTE** | 9 | 9 |
| **VTW** | 6 | 8 |
| **VTE** | 9 | 10 |

**Table 5.1: Number of participants and model-runs in the different experimental groups**

Table 5.1 shows the number of participants and model runs for each experimental group. Each model run is carried out using the same set of rules and ACT-R parameters values; the variability obtained is a product of particular paths of actions followed during each run that reflect different rewards from the environment and the runtime stochastic component of ACT-R. A full model run lasts 6,240 seconds (24 sessions of 260 seconds).

## 5.2 Assessing the QOF

A common way of determining the QOF of a model is by using performance and latency comparisons (Lebiere et al., 2001). Kiefer & Urbas (2006) used the average number of patterns completed per minute and the average reaction time in ms. per pattern. Veksler, Gray, & Schoelles (2007) using the *Table Decision* task (section 2.2.3.4) considered three measures of performance: the average trial duration, the number of mouse clicks in the different cells and the inter-cell click interval (the time spent between cell clicks). In the model of Lee & Taatgen (2002) the model was compared at three levels: score per trial, time per unit task, and individual keystrokes. The CMU-ASP model (Taatgen, 2005) was evaluated by considering the number of aircraft successfully classified in each scenario, and the average time per scenario to perform each one of the three subtasks (select an aircraft, classify an aircraft and enter the classification). In the model of Peebles & Bothell (2004) of the RT task, the mean number of responses and the mean response time were used for comparing the quality of fit. Jones, Ritter & Wood (2000) identified nine measures of behaviour in the Tower task (section 6.1.3.1). They argue that using a variety of measures provides a more detailed match. Two of these measures are the time taken to complete the tower and the number of constructions made in completing it. Jones et al. (2000) used aggregated data for determining the QOF: they compared the aggregated data from 10 runs of the model with the performance of 5 participants. All the metrics are obtained by trial and can be averaged over training programme, participant, or test condition. To show the QOF modellers are accustomed to presenting a graph comparing the data from participants and the model and showing that the tendency is captured (Taatgen, 2005); this practice is used throughout the chapter.

In the following sections correlation data is provided to demonstrate the QOF; nevertheless, given that microworlds do not allow full experimental control (section 2.3.5), there are several section where the existence of meaningful interactions for both participants and the model are demonstrated instead by using ANOVA tests.

### 5.2.1 General performance

The first paragraph in section 2.3.1.5 describes how performance in measured in FireChief.

#### 5.2.1.1 Training phase

Considering performance in the training phase the value the QOF is good ($r^2$=.907, RMSD=.0773). To assess the impact of practice, data extracted from the model for CT performance for the first four trials was compared with performance for the last four trials. A one-tailed one-way within-subjects ANOVA was performed to determine whether there was a significant learning effect, the data are approximately normally distributed and the discrepancies are not too wide in the data generated by the different groups of trials. Results show that, as in the case with the human data (see section 3.4.4), there is a significant effect of learning ($F(1,38)=4.41$, $p<.05$). Figure 5.1 shows the average performance for participants and the model. The model offers an explicit explanation of how this improvement in performance is achieved, by means of better strategy selection based on the feedback provided at the end of the trial, sensitivity to wind strength, and a better implementation of strategies as a consequence of tuning the utility of productions to the CT trial. Figure 5.2 shows the data for the VT; as in the case of the human data, there is a significant interaction between task complexity and performance in the data generated by the model in the VT programme ($F(1, 34) = 4668.1$, $p<.001$).

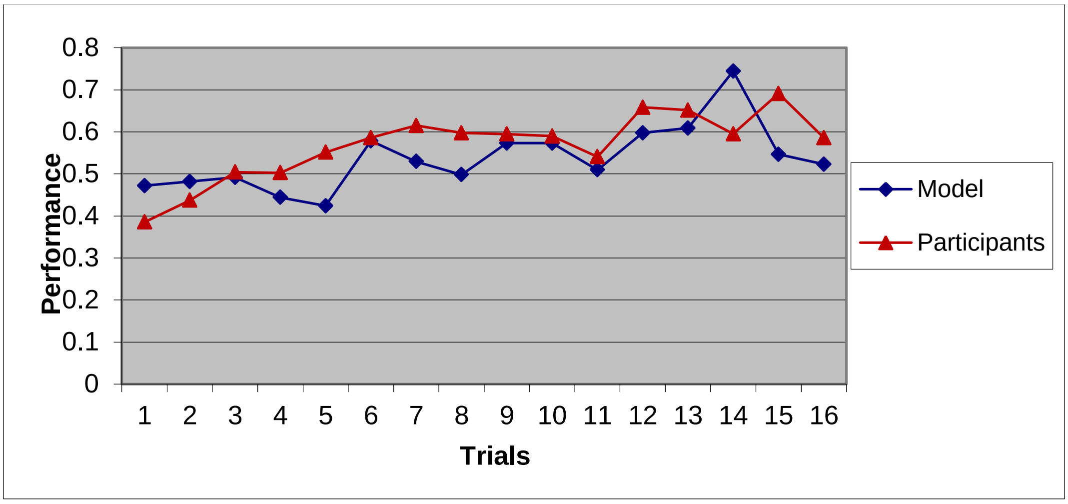



**Figure 5.1: Performance comparison by trials in the CT**

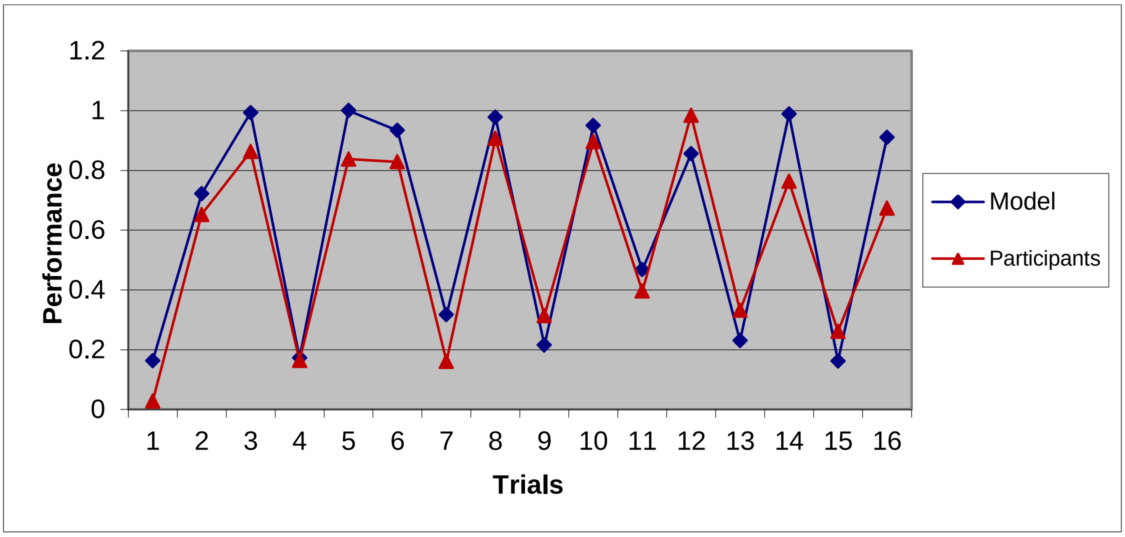



**Figure 5.2: Performance comparison by trials in the VT**

### 5.2.1.2 Testing phase

Table 5.2 shows the performance comparison for the test phase. Performance differences in the VTW condition are a result of the model's lack of ability to execute the Stop strategy with the same success as participants. The model also shows lower performance than participants in the CTE condition. In general terms the model has a preference for the Barrier strategy when the wind changes direction while participants choose both Barrier and Stop strategies.

Both participants and the model learn to use the Barrier strategy when appliance efficiency is reduced. Learning effects during the test phase were not studied as the topic of interest was cognitive inflexibility which is further discussed in sections 5.2.2.3 and 6.1.2.4. As this research is primarily concerned with strategy use and its implications for performance in dynamic situations strategy use and its relation with performance is discussed in the following sections, including how strategies change within trials.

**Testing Phase**

| | Participants | Model |
|---|---|---|
| CTW | 78.81 | 79.44 |
| CTE | 78.70 | 74.03 |
| VTW | 83.61 | 78.28 |
| VTE | 76.13 | 70.58 |

**Table 5.2: Comparison during testing phase**

### 5.2.2 Strategy use: performance and frequency

Table 5.3 shows the comparison of human study data with the model data for performance (top) and frequency (bottom) of the four strategies at the aggregate level. Because the modelled data present high variability, metric averages are used for establishing the QOF; other researchers have used a similar approach (Anderson & Lebiere, 1998).

**PERFORMANCE**

| Training | CT | | VT | |
|---|---|---|---|---|
| | Data | Model | Data | Model |
| Barrier | 81.225 | *80.865 | 75.625 | *75.69 |
| NonBarrier | 73.445 | *71.54 | 58.115 | 83.22 |
| Stop | 92.995 | 70.48 | 94.345 | *83.68 |
| Follow | 63.6 | *65.845 | 57.96 | 84.665 |

**(a)**

| Test | CTW | | CTE | | VTW | | VTE | |
|---|---|---|---|---|---|---|---|---|
| | Data | Model | Data | Model | Data | Model | Data | Model |
| Barrier | 79.21 | *78.24 | 86.53 | *76.64 | 81.67 | *79.43 | 80.95 | *71.71 |
| NonBarrier | 63.56 | 71.28 | 72.74 | 64.63 | 74.21 | *70.57 | 60.41 | *65.2 |
| Stop | 81.89 | *81.15 | 0 | 0 | 89.78 | *83.27 | 48.95 | *59.76 |
| Follow | 73.29 | 78.53 | 46.4 | 67.45 | 66.09 | 75.01 | 65.67 | 0 |

**(b)**

**FREQUENCY**

| Training | CT | | VT | |
|---|---|---|---|---|
| | Data | Model | Data | Model |
| Barrier | 0.655 | *0.55 | 0.29 | *0.3 |
| NonBarrier | 0.175 | *0.2 | 0.18 | *0.195 |
| Stop | 0.065 | 0.16 | 0.42 | *0.425 |
| Follow | 0.11 | *0.095 | 0.11 | *0.085 |

**(c)**

| Test | CTW | | CTE | | VTW | | VTE | |
|---|---|---|---|---|---|---|---|---|
| | Data | Model | Data | Model | Data | Model | Data | Model |
| Barrier | 0.59 | *0.69 | 0.77 | *0.86 | 0.34 | 0.52 | 0.72 | *0.71 |
| NonBarrier | 0.15 | 0.06 | 0.16 | *0.13 | 0.11 | *0.13 | 0.16 | *0.16 |
| Stop | 0.15 | 0.1 | 0 | 0 | 0.48 | 0.25 | 0.04 | 0.13 |
| Follow | 0.11 | *0.15 | 0.07 | 0.01 | 0.07 | *0.11 | 0.07 | 0 |

**(d)**

**Table 5.3: Performance in the training (a) and test (b) phases and frequency in the training (c) and test (d) phases considering strategy use. An asterisk represents good correlation.**

### 5.2.2.1 Constant Training condition

Considering participant data for the training phase, a two-tailed one-way within-subjects ANOVA revealed that there is a significant interaction between strategy use and performance in the CT ($F(3, 260)=52.75$, $p<.001$) and in the VT ($F(3,218)=84.48$, $p<.001$). In the same way, considering the data generated with the model, there is a significant interaction between strategy use and performance in the CT ($F(3,300)=24.23$, $p<.001$) and in the VT ($F(3,284)=3.31$, $p<.05$). This result means that strategy use has a significant impact on performance during the training phase and that this interaction is captured by the model. This is particularly relevant as this study is mainly interested in strategy use. If performance of the few trials were the Stop strategy was successfully implemented in the CT condition is not taken into account, the QOF of the model for the training phase is good ($r^2=.975$, RMSD=1.88). Participants have a preference for using the Barrier strategy as this strategy usually generates good performance and the model successfully captures this tendency both in frequency and performance level. The NonBarrier strategy occupies second place considering frequency and again the model captures this tendency. Albeit not used frequently, the Follow strategy is the worst strategy for both the model and participants. Finally, the Stop strategy is seldom used in this condition and participants implement this strategy more successfully as two of them were able to implement the Stop strategy in such a fashion that the fire was stopped without using CF commands in various CT trials. These two participants were not deemed as outliers because they adapted their strategies during the testing phase. It is worth mentioning that the model is able to replicate this behaviour by exerting a high amount of control over the way actions are executed but, left to itself, the model does not have enough time to learn an adequate sequence of actions because its early attempts at using the Stop strategy will lead to poor performance and hence to rejection of the strategy. In other words, the tenacity to hang on to a preferred strategy (such as Stop) even under adverse conditions (the CT programme) is not captured by the model.

### 5.2.2.2 Variable Training condition

Considering the VT programme the amount of times the Follow strategy is used is very low: 11% for participants and 8.5 % for the model (see table 5.3 (c)). Participants achieve better performance than the model as some participants selected the Follow strategy in very easy trials. The model is not able to capture the low performance obtained by participants when executing the NonBarrier strategy. Considering the most structured strategies, Stop and Barrier, the correlation between the model and the data is good ($r^2$=.987, RMSD=5.37). The high performance obtained by the model using the NonBarrier strategy can be explained by differences in the complexity of trials: if the model chooses to execute the NonBarrier strategy in a very easy trial (such as trials 2, 6, 8 or 12) it will obtain good performance. The fit of the model to how the other strategies are executed is quite good: the RMSD for the Barrier strategy is .18. Table 5.3 also shows the frequency of strategy execution. Note that during the CT the Stop strategy is executed a few times whilst the Barrier strategy is the most popular; for this reason the overall fit of the model remains high regardless if its lack of ability to perform the Stop strategy as well as participants. Considering all strategies the RMSD is high: 16.06. If we obtain the RMSD for the most structured strategies Stop and Barrier the result is 5.3 and if we get the RMSD for the Barrier strategy the result is 0.315. This result shows that the model is accurate when executing structured strategies under the VT programme training conditions.

At this point is possible to identify a couple of strategy interactions involving the most structured strategies that will be relevant during the testing phase. (1) The Barrier strategy does not yield the same level of performance as in the CT programme: it is reduced by 7% on average for both participants and the model; and (2) the Stop strategy is implemented with high success and both the model and participants use it more frequently than in the CT condition: 42% and 42.5% respectively. The second interaction can occur because, in contrast with the CT condition, the VT condition allows the successful implementation of the Stop strategy in several trials.

### 5.2.2.3 Testing phase condition

Considering the human data in the testing phase, a two-tailed one-way within-subjects ANOVA revealed that there is a significant interaction between strategy use and performance in the CTW group ($F(3,42)=3.812$, $p<.05$), in the CTE group ($F(2,67)=21.796$, $p<.001$), in the VTW group ($F(3,40)=12.361$, $p<.001$), and in the VTE group ($F(3,65)=11.002$, $p<.001$). This means that the strategy selection has a significant impact on performance also during the testing phase. Considering the data generated with the model, there is no significant interaction between strategy use and performance in the CTW group ($F(3,76)=1.835$, $p=.148$), but the CTE group shows a significant difference ($F(1,70)=3.425$, $p<.05$) between the use of the Barrier strategy and the other strategies. There is a significant difference in the VTW group ($F(3,60)=4.500$, $p<.05$), and in the VTE group ($F(2,77)=9.234$, $p<.001$). The fit of the model for the testing phase is good ($r^2$=.926, RMSD=8.1). As the topic of interest during the testing phase is cognitive inflexibility specific comments related to this phenomenon are discussed in the following subsections.

### 5.2.2.3.1 Condition CTW

It is noticeable that both the model and participants have a preference for using the Barrier strategy at the moment in which the wind changes direction. Nevertheless the most interesting effect is how the use of the CF command is disrupted as a consequence of wind direction changes. The model is able to capture how performance decreases between trials 16 and 17 for strategies that use CF commands ($R^2$= 0.796). Although it is harder to implement the Barrier and NonBarrier strategies both participants and the model keep using them. This is considered evidence of cognitive inflexibility as the Stop strategy is more suitable for dealing with a change in wind direction. To create a barrier it is compulsory to know the direction in which the fire is going to develop, and this information is not known when wind changes direction so often.

### 5.2.2.3.2 Condition CTE

The most interesting interaction related to cognitive inflexibility in this experimental condition is how participants that learnt how to use the Barrier strategy during the training phase are not affected by this environmental change. A one-tailed one-way within-subjects ANOVA was performed to determine whether there was a significant performance difference in using the Barrier strategy between the 16$^{th}$ and 17$^{th}$ trial and the result is that for both participants and the model there is no significant difference (participants $F(1,7)=0.45$, $p=.45$; and for model ($F(1,7)=1.3$ $p=.33$). Using a strategy different from Barrier results in poor performance due to the fact that DW commands prove less effective in this experimental condition. The data in table 5.3 (d) shows that Only-DW strategies are almost abandoned by the CTE group during testing. This phenomenon can be explained by two mechanisms. (1) Participants in the CT group are well trained in the creation of barriers during training so presumably they will try to create a barrier when the 17$^{th}$ trial starts (unbeknownst to them that the efficiency of their appliances has been reduced). As far as these participants are concerned, the creation of the barrier may stop the development of the fire and, when they notice that DW commands are less effective, only then would they try to use DW commands in a different fashion; but the fire would in any case be halted by the barrier. In this particular circumstance being inflexible in strategy use obtains good results. (2) When participants try to execute a DW command over a strong fire, they will receive an alarm and the DW won't be completed; eventually participants must start issuing CF commands in order to stop the fire.

### 5.2.2.3.3 Condition VTW

The most important interaction related to cognitive inflexibility is that, when wind changes direction, both participants and the model that underwent the VT should execute the Stop strategy more frequently than their counterparts in the CTW condition. By comparing conditions CTW and VTW it is possible to see that the frequency of use of the Stop strategy is higher in the VTW condition for both participants and the model (see table 5.3-d). A one-tailed one-way between-subjects ANOVA was carried out to test whether using the Stop strategy in the 17$^{th}$ trial by VTW participants results in significant better performance that using Barrier in the same trial by CTW participants. The results shows that both for participants ($F(1,15)=14.62$ $p<.05$) and the model ($F(1,15)=8.35$ $p<.05$) better performance is obtained by implementing the Stop strategy when wind changes direction. The result is that the VTW group performs better than the CTW group and therefore there is a proven advantage of being flexible when wind changes direction.

##### 5.2.2.3.4 Condition VTE

Both participants and the model have a significant preference for the Stop strategy during the training phase but are able to switch to the more successful Barrier strategy when a change in the environment takes place (remember that Only-DW strategies do not work in this condition). What is relevant to the cognitive inflexibility phenomenon is to determine whether participants in the CTE group show better performance using the Barrier strategy than the ones in the VTE group. A one-tailed one-way between-subjects ANOVA was carried out to test whether performance using the Barrier strategy is better for participants in the CTE group during the first four trials of the test phase in comparison to participants in the VTE group. The first four trials were selected as performance becomes stable for the VTE group after the fifth test trial. Results show that performance using the Barrier strategy is better in the CTE compared to the VTE both for participants ($F(1,7) = 6.96$ $p>.05$) and the model ($F(1,7)=10.4$, $p>.05$). The model provides an explanation of this phenomenon in terms of strategy consolidation (see section 6.1.2.2).

#### 5.2.2.4 Within-trial strategy change

| | Participants | | | Model | | |
|---|---|---|---|---|---|---|
| **Group** | **Training** | **Testing** | **Overall** | **Training** | **Testing** | **Overall** |
| **CTW** | 6.00 | 4.88 | 10.88 | 4.90 | 1.90 | 6.80 |
| **CTE** | 6.33 | 2.44 | 8.78 | 5.22 | 1.56 | 6.78 |
| **VTW** | 11.67 | 3.00 | 14.67 | 7.25 | 2.63 | 9.88 |
| **VTE** | 11.44 | 3.78 | 15.22 | 7.40 | 1.90 | 9.30 |

**Table 5.4: frequency of within-trial strategy change**

Within-trial strategy change refers to the number of times a strategy is changed during the two phases of the experiment. Table 5.4 shows the average number of times participants and the model change strategy during the training and testing phases ($r^2$=.93 RMSD=1.43). Overall, in the training phase participants change strategies 6.2 times in the CT programme while the model changes strategies 5.1 times. In the VT programme participants change strategies 11.55 while the model changes strategies 7.3 times (The RMSD for the training phase is 1.56 and the RMSD for the test phase is .99). There is a significant difference in the frequency of strategy change both for participants ($F(1,30) = 35.692$, $p<.001$) and the model ($F(1,35)=19.354$, $p<.001$). Nevertheless participants change strategy with more frequency than the model, particularly in the VT programme. Both participants and the model change strategy more often in the CTW condition in comparison with the CTE condition. Nevertheless the model changes strategy with higher frequency in the VTW condition. This last result is also an indication that the model is having problems dealing with changes in wind direction.

### 5.2.3 A closer look at strategy use: command use

Strategy execution depends upon how commands are issued. As previously mentioned the lack of full experimental control makes it impossible to deliver the same experimental conditions for all participants/model runs. For this reason, rather than providing an analysis based on command frequency, results focused on functional interactions between command use, strategy execution and task performance are given instead. The objective of this section is to demonstrate how the model is able to capture participants' behaviour at the level of command use. This section is relevant as there are aspects of the data that are masked when strategy use is considered only. For instance, a one-tailed one-way between-subjects ANOVA was carried out to test whether there is a significant performance difference between the CTE and VTE groups when implementing the Barrier strategy during training. The result was that there is no significant difference between conditions for participants ($F(1,102)=1.067$, $p=.304$) and the model ($F(1,117)=.386$, $p=.536$). Nevertheless, by analysing the utility of productions, it can be observed that the CT and the VT conditions generate a different set of utility values in productions by the end of the training, and these different sets represent different ways of implementing the Barrier strategy (section 5.1.2.3). This is the kind of insight provided in this section.

The model is able to replicate the pattern of command use frequency in the CT programme ($r^2$=.98 RMSD=5.04) and in the VT programme ($r^2$=.94 RMSD=8.02). The most notable difference in the pattern of command use during training is that the execution of the Stop strategy for the VT involves a lower number of DW commands in comparison to the CT for both participants and the model, a phenomenon related to the second interaction described in section 5.2.2.2. During the test phase, for the CTW test condition, the model uses fewer commands than participants (as participants perform more repetitions of commands) but it captures the tendency of the data ($r^2$=.975 RMSD=5.67). The tendency of the data is also captured in the VTW group ($r^2$=.982 RMSD=6.3). Participants in the CTE group during the test phase use more commands than the model ($r^2$=.982 RMSD=9.09). This can be an indication of the difficulty associated with dealing with a reduction in the efficiency of appliances. Because DW commands are executed mainly by copters, the 'chaos' generated by this change in the environment is associated more with the use of copters than with trucks. Finally the model also captures the pattern of command use for group VTE ($r^2$=.949 RMSD=12.49). The most relevant finding provided by the use of transition matrices is that the pattern of command transitions for the Barrier strategy is the same in the CT and VT programmes for both participants and the model. This means that the consolidation of the Barrier strategy is mainly related to the spatial distribution of commands rather than to the transition between commands (see section 5.1.2.3). Now that the QOF of the model in relation to overall command use frequency has been summarized it is time to explore the most relevant findings.

### 5.2.3.1 Latency of the Control Fire command: Barrier vs. NonBarrier

Table 5.5 shows the latencies related to CF command use (only successful CF are considered) for participants and the model, distinguishing between the Barrier strategy "BA" and the NonBarrier strategy "NBA". Overall the model is able to replicate the tendency observed in the empirical data (CT condition $r^2$= .997 RMSD=.61, VT condition $r^2$=.837 RMSD=1.1, CTW group $r^2$=.919 RMSD=1.7, VTW group $r^2$=.952 RMSD=.59, CTE group $r^2$=.895 RMSD=.63 and VTE group $r^2$=.898 RMSD=.65).

| | CT | | VT | | CTW | |
|---|---|---|---|---|---|---|
| **Type** | **Data** | **Model** | **Data** | **Model** | **Data** | **Model** |
| BA/barr | 7.7 | 4.0 | 8.4 | 4.5 | 6.6 | 4.0 |
| NBA/barr | 8.4 | 4.7 | 9.5 | 4.1 | 6.6 | 5.0 |
| BA/-barr | 11.4 | 6.7 | 13.5 | 6.6 | 12.3 | 6.3 |
| NBA/-barr | 12.2 | 7.1 | 12.0 | 7.5 | 12.0 | 7.0 |

| | VTW | | CTE | | VTE | |
|---|---|---|---|---|---|---|
| **Type** | **Data** | **Model** | **Data** | **Model** | **Data** | **Model** |
| BA/barr | 7.8 | 3.9 | 7.1 | 3.9 | 7.6 | 4.2 |
| NBA/barr | 6.8 | 3.8 | 7.4 | 5.4 | 8.6 | 3.8 |
| BA/-barr | 9.7 | 6.5 | 9.4 | 5.9 | 10.9 | 6.4 |
| NBA/-barr | 11.4 | 6.9 | 10.4 | 6.8 | 10.7 | 7.4 |

**Table 5.5: Mean duration of CF commands (BA = Barrier, NBA = NonBarrier, the second term in type indicates if the CF belongs to a barrier or not, so BA/barr refers to CF issued under the Barrier strategy which are part of the barrier).**

Considering all trials, both model and human study (section 3.4.5.2), there is a significant difference in latency between the execution of a CF belonging to a barrier than in a CF that does not belong to a barrier ($F(1,737)=200.373$, $p<.001$). In the CT condition, for both the human study and the model, a CF command take the least time when it belongs to a barrier and is executed under the Barrier strategy and takes the longest time when it does not belong to a barrier and is executed under the NonBarrier strategy. The model is able to capture this tendency, but falls short regarding absolute times, that is, the model is always faster than participants. The human data in the VT condition shows a similar pattern. During the test phase the latency of CF commands issued by participants that do not belong to a barrier are longer than those that do belong to a barrier in all conditions, and the same was observed in the model. As mentioned before a key factor is that the mouse pointer has to be moved longer distances for CF commands that do not belong to a barrier, but latency differences are also a result of the different cognitive and perceptual actions involved in adding a segment to a barrier in comparison to placing an isolated CF, particularly the fact that the model tends to keep its attention on the unit that is going to execute the CF commands when a barrier is being constructed.

### 5.2.3.2 Latency of the Drop Water command: copters vs. trucks

Only completed DW commands are used for comparing command duration. In table 5.6 the Only-DW strategies Stop and Follow are considered. Overall the model is able to replicate the tendency observed in the empirical data (CT condition $r^2=.179$ RMSD=.77, VT condition $r^2=.892$ RMSD=.73, CTW group $r^2=.96$ RMSD=.58, VTW group $r^2=.883$ RMSD=.1.8, CTE group $r^2=.999$ RMSD=.95, the model never executed the follow strategy in the VTE condition so no QOF of data was obtained). Note that the highest RMSD was obtained for the VTE test condition mainly due to the fact that the model never executed the Follow strategy during the test phase whilst participants executed this strategy 5 times. There is a significant difference between a DW command issued by a copter and a DW command issued by a truck, regardless of the strategy used. In general terms it takes less time for participants and the model to start a DW command if a copter is used. These differences in time are related to different ways in which units are used. Because copters are able to extinguish fires of higher intensity it is a good practice to use them to drop water over stronger fires.

| CT | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 6.75 | 7.5 | 7.89 | 8.29 |
| Copter | 5.03125 | 6.63158 | 5.76 | 5.76 |

| VTW | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 8 | 9.3 | 7.66 | 7.66 |
| Copter | 5.7 | 6.7 | 6.10 | 5.94 |

| VT | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 9.17442 | 10.24 | 8.46 | 8.46 |
| Copter | 7.77907 | 7.16 | 6.59 | 6.96 |

| CTE | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 0 | 7.2 | 0.00 | 11.67 |
| Copter | 0 | 4.8 | 0.00 | 7.44 |

| CTW | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 9 | 9 | 7.37 | 7.30 |
| Copter | 5.6 | 6 | 6.14 | 5.77 |

| VTE | Participant | | Model | |
|---|---|---|---|---|
| | Stop | Follow | Stop | Follow |
| Truck | 4 | 8.6 | 8.48 | 0.00 |
| Copter | 5.7 | 6 | 7.17 | 0.00 |

**Table 5.6: Mean duration of DW commands**

Because the location of stronger fires is contingent upon the direction in which the wind is blowing these fires tend to create clusters. In this way copters used when executing either the Stop or Follow strategies are usually close to each other: they operate in the area currently occupied by the fire-front. On the other hand, trucks attacking the fire have a bigger operation area, which can explain why it takes longer to initiate DW commands with this kind of unit. The fact that the model was able to replicate the latencies observed in human data suggests that strategies are well captured by the model. The model is also able to capture the increment in the time required to execute a DW command between the VT and CT conditions observed in the human data. This increment is caused by the hardest VT trials, where fast fire development tends to cause a wider dispersion of units. As observed in the empirical data, the model shows a significant difference between using trucks and copters to execute DW commands: DW commands issued by copters take less time than DW commands issued by trucks ($F(1,888)=27.02$ $p<.001$). This result demonstrates that both participants and the model are using copters and trucks a similar fashion. The CTE and VTE conditions are not discussed because the majority of DW commands are cancelled due to the reduction in the efficiency of appliances and there is therefore insufficient data for comparison.

Table 5.7 shows the utility of productions related to attacking the fire after training is completed. Section 4.2.3 describes how the utility of productions can be used to understand the model’s behaviour. Subsection 9 in section 4.3.1 describes the three modes of attacking the fire. In both training conditions the model has a low preference for attacking weak fires, but under the CT condition this aversion is higher: due to the high wind strength the model should attack weak fires only if the strong ones are under control; on the other hand easy trials in the VT programme allow the model to attack weak fires without high risk. In both training conditions the model found that attacking strong fires is rewarded.

| | Trials | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Attack fire** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Attack Strong CT | 0.67 | 0.85 | 0.21 | 1.15 | 1.24 | -0.96 | 0.18 | -0.17 | -0.59 | 0.39 | 1.17 | -1.72 | 0.81 | 0.83 | 1.07 | 0.94 |
| Attack Strong VT | 1.11 | 1.15 | 0.34 | 0.78 | 1.37 | 1.18 | 0.67 | -0.16 | 0.71 | 2.06 | 2.83 | 1.37 | 1.29 | 0.21 | 0.62 | 0.45 |
| Attack Strongest CT | 0.00 | 0.00 | 0.14 | 0.23 | 0.23 | 0.30 | 0.35 | 0.40 | 0.44 | 0.46 | 0.49 | -1.36 | -1.36 | -1.36 | -1.36 | -1.36 |
| Attack Strongest VT | 0.25 | 0.27 | 0.28 | 0.37 | 0.37 | 0.40 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 | 0.42 | 0.76 | 0.70 | 0.61 | 0.60 |
| Attack Weak CT | 0.00 | 0.33 | 0.37 | -0.54 | 0.65 | 0.77 | 0.84 | 1.08 | 0.07 | -0.18 | -2.27 | -2.27 | -1.38 | -1.43 | -2.46 | -2.29 |
| Attack Weak VT | -0.52 | -0.52 | -0.60 | -0.60 | -0.60 | 0.02 | 0.50 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | -0.37 | -0.37 | -0.04 | -0.09 |

**Table 5.7: Utility values associated with attacking fire**

The most interesting difference related to the use of the DW command comes from attacking the strongest fires: under the VT condition this behaviour is considered useful while in the CT condition there is an aversion to this way of attacking the fire. This is a consequence of the lack of use of the Stop strategy in the CT condition (because the strongest fires exceed the capacity of the fire-fighting units) and the good feedback that the model receives when using the Stop strategy in the VT condition (due to the existence of easy trials). This is also an indication of the lack of practice of the Stop strategy by the model in the CT condition.

#### 5.2.3.3 Use of the Move command: strategy consolidation

A closer analysis of the use of the Move command reveals an interesting interaction related to strategy consolidation. A one-tailed one-way between-subjects ANOVA was performed to determine whether there are more short movements (of length equal to one cell, see section 3.4.5.4) when executing the Barrier strategy in the CT in comparison to the VT. The result show that this is the case both for participants ($F(1,30)=14.88$ $p<.001$) and the model ($F(1,30)= 13.05$ $p<.001$). When participants and the model consolidate the Barrier strategy they are able to create a barrier efficiently; this means that participants tend to issue CF commands in close proximity to each other, and the shortest distance between two cells is the length of one cell. Section 3.4.4 shows how better performance during the Test phase is obtained when the Barrier strategy is consolidated and section 6.1.2.3 explores the consolidation of the Barrier strategy in detail.

#### 5.2.3.4 Move and Control Fire interaction

| | **Participant** | **Model** |
|---|---|---|
| **CF in Barrier** | 13.12 | 15.17 |
| **CF not in Barrier** | 16.57 | 16.24 |
| **First CF Barrier cell** | 20.86 | 22.01 |

**Table 5.8: Comparison of types of moves**

Table 5.11 shows the average time required for executing a *Move* command before issuing a *CF* command (based on whether a CF belongs to a barrier, does not belong to a barrier, or starts a barrier) following the Barrier strategy. The model captures the tendency discussed in section 3.4.5.2: CF commands issued in cells outside the barrier take more time that CF commands issued for a barrier. The second important tendency observed in the data is that the time required to execute the very first movement before issuing a CF for a barrier requires more time than the other types of movements (participants, $F(415,2)= 28.03$, $p<0.001$; model, $F(562,2)= 109.08$, $p<-001$). Chapter 4 (section 4.2.4.2) describes the different processes that are executed before issuing the first CF of a barrier.

### 5.2.4 Evidence of adaptivity

Several adaptive behaviours identified during the data analysis phase are replicated by the model.

#### 5.2.4.1 Strategy adaptivity

In the CT condition both participants and the model use the Barrier strategy increasingly more frequently until trial 16. Figure 5.3 shows the proportion of times the Barrier strategy is used in the CT condition. Note that both for participants and the model the Barrier strategy is used with more frequency as more trials are completed and that there is a drop in the use of the Barrier strategy after Trial 16 (the beginning of the test phase). This phenomenon is expected due to the change in conditions that renders the Barrier strategy inappropriate for the test trials (section 3.1.3.2). The rest of this section discusses strategy adaptivity during the training phase; the impact of strategy selection during the training phase is discussed in the cognitive inflexibility section (6.1.2.4). Table 5.9 shows the average utility values of the rule set associated with strategy choice (Section 4.2.3.2 explains how the utility of these rules is modified). A utility value is high or low depending on the utility values of its competing productions. The CT run leads the model to adopt a clear preference for the Barrier strategy. Table 5.10 shows the selection of strategies and the resulting performance for the corresponding model runs. This table only shows the strategy selected at the start of each trial, remember that the model is able to switch strategies within trials. The choice of strategies in 5.10 is based on the utilities shown in table 5.9. Note that the utility of selecting the Barrier strategy drops to 1.17 in trial number eight (cell with asterisk in table 5.9). This drop occurs due to the low performance reported in table 5.14 (cell with asterisk in table 5.10); the seventh trial yielded a result of 69.8 (a poor result compared with previous trials) which lowered the perceived utility of the Barrier strategy from the model’s perspective.

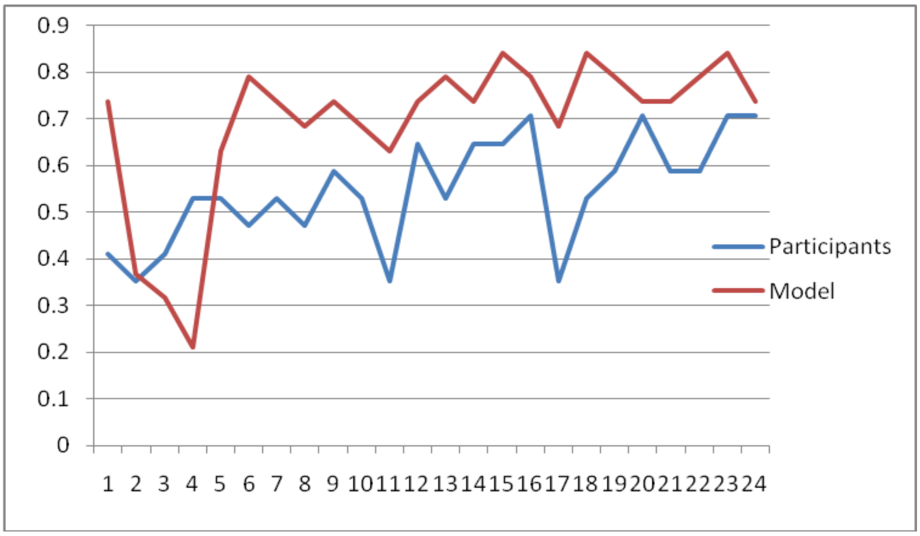



**Figure 5.3: Use of Barrier strategy in the CT condition considering all trials**

The model run for the VT condition changes strategies more frequently than the model run for the CT condition. Table 5.10 shows that the first strategy selected by the model in the VT condition run is Stop which yielded very low performance (cell with double asterisk in table 5.10). As can be seen at the end of the first trial the utility associated with the use of the Stop strategy is –12.38 (cell with double asterisk in table 5.9). In the next three trials the strategy used was Barrier which is abandoned after obtaining low performance in trial 4 (as indicated by the unchanging values over trials 4 to 6 in table 5.9). The NonBarrier strategy is selected next and yields very high performance due to the low complexity of the trial. The NonBarrier strategy drops its utility in the seventh trial due to the low performance obtained but it is selected again in the following trial where it obtains a high performance. In trial 9 the Barrier strategy is selected again but the result is not good enough so in the next trial the Stop strategy is selected and remains in use until trial 13 (the hardest VT trial scenario).

| | Trials | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Select Strategy** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Select Barrier CT | 2.00 | 2.52 | 9.56 | 7.51 | 9.51 | 10.00 | 4.18 | *1.17 | 8.70 | 0.52 | 7.42 | 10.00 | 10.00 | 7.90 | 10.00 | 6.04 |
| Select Barrier VT | 2.00 | 8.38 | 10.00 | 4.31 | 4.31 | 4.31 | 6.31 | 6.31 | 7.65 | 7.65 | 9.65 | 9.65 | 10.00 | 10.00 | -1.52 | -1.52 |
| Select Follow CT | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Select Follow VT | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Select NonBarrier CT | 0.00 | 0.00 | 0.00 | 0.00 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 |
| Select NonBarrier VT | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 3.02 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Select Stop CT | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 | -4.27 |
| Select Stop VT | **-12.38 | -9.38 | -6.38 | -6.38 | -3.38 | -0.38 | -0.38 | 2.62 | 2.62 | 9.67 | 9.10 | 10.00 | -3.63 | -0.63 | -0.63 | -0.63 |

**Table 5.9: Utility values associated with strategy use**

The VT condition elicits a different pattern of behaviour. What impacts strategy selection the most are the significant variations in complexity existing in the VT programme. Some trials are very easy and generate good feedback regardless of strategy selection. This could be misleading both to participants and the model. In the case of the model it receives positive feedback for an arbitrary choice of strategy that serves to increase the utility of a strategy that might not be a good choice for harder trials. Some trials are so hard that only structured strategies such as Barrier and Stop can produce good results, but if these strategies are not selected the model cannot exercise them. By having these extreme changes in complexity between trials it is possible for the model to develop a preference for less effective strategies, such as in the case of the NonBarrier strategy which is used in trials 5 to 8 after obtaining 98.24 in a very easy trial.

This section described how the Barrier strategy becomes the preferred one during the training phase. This phenomenon produced cognitive inflexibility during the test phase, which can also be traced to the utility of productions; this is discussed in sections 6.1.2.3 and 6.1.2.4.

| | CT | | VT | |
|---|---|---|---|---|
| **Trial** | **Perf.** | **Strategy** | **Perf.** | **Strategy** |
| 1 | 73.56 | STOP | **49.23 | STOP |
| 2 | 76.15 | BARRIER | 99.06 | BARRIER |
| 3 | 93.18 | BARRIER | 100 | BARRIER |
| 4 | 78.03 | BARRIER | 71.29 | BARRIER |
| 5 | 71.8 | NONBARRIER | 100 | NONBARRIER |
| 6 | 95.53 | BARRIER | 98.24 | NONBARRIER |
| 7 | *69.8 | BARRIER | 57.38 | NONBARRIER |
| 8 | 70.98 | BARRIER | 100 | NONBARRIER |
| 9 | 94.48 | BARRIER | 81.16 | BARRIER |
| 10 | 60.87 | BARRIER | 96.16 | STOP |
| 11 | 92.6 | BARRIER | 87.31 | STOP |
| 12 | 94.95 | BARRIER | 93.3 | STOP |
| 13 | 90.72 | BARRIER | 47.37 | STOP |
| 14 | 78.14 | BARRIER | 100 | BARRIER |
| 15 | 91.3 | BARRIER | 46.28 | BARRIER |
| 16 | 73.56 | BARRIER | 92.08 | NONBARRIER |

**Table 5.10: Strategy use for two runs of the model**

### 5.2.4.2 Refilling the tank and performance

At first appearance it might seem as if the way to achieve best performance might be to refill the tank as soon as it becomes empty, however, data from both the model and human participants shows that both the model and participants that refill units as soon as the tank is depleted (the adaptive participants) are not better performers. A one-tailed one-way between-subjects ANOVA revealed that there is not a significant difference in performance for either participants ($F(1,21)=.01$, $p=.921$) or the model ($F(1,21)=1.6$, $p=.22$). The model keeps track of the number of DW commands that have been issued, but the rule that sends a unit to the dam to refill its tank must win a competition with other rules. What is observed is that the rule that can send a unit to refill as soon as its tank is depleted is not favoured over the rest, so there is no priority in executing this action. Participants do not seem to prioritize this task either. It was observed that the model prioritizes actions such as sending copters to locations in which fires can be attacked or perceptual actions that sense the state of the fire. This result suggests that it is cognitively cheaper to wait for an "empty tank" alarm and send the unit to a nearby dam (the vast majority of trials have several dams) than to keep checking the level of water.

#### 5.2.4.3 Waiting behaviour

Imagine that you are executing a Move command using a truck. You just completed the drag-and-drop mouse command so the truck is disabled by FireChief and starts its movement towards the target cell. What will you do next? You can wait until the truck arrives at the target cell, switch attention to another unit, check the status of the fire, check the strength of the wind, and so on. When the model finds itself in a similar situation it must also decide what single action to execute next from the several options available. An adaptive behaviour observed both in participants and in the model is the emergence of an interaction between the length of the movement of units and waiting behaviour. This phenomenon was not detected during the analysis of participant data, a hint towards this discovery occurred during the movement length analysis for the model. To be more precise: if the movement's length is shorter than 2 cells, the model tends to wait for the unit to arrive; otherwise it selects another option, such as moving its attention to another unit. This interaction effect is mediated by the type of command issued after the movement: the model favours waiting when executing CF commands, but not DW commands. Figure 5.4 show the proportion of times that waiting behaviour is observed considering the Move→CF and Move→DW sequences for participants and the model. The figure shows that waiting behaviour is favoured when CF commands are involved in the sequence. There is a significant difference in waiting behaviour between Move-CF and Move-DW sequences in participants ($F(1,46)=184.210$, $p<.001$) and the model ($F(1,46)=88.734$, $p<.001$). Section 5.2.1 shows that as trials are completed both participants and the model execute the Barrier strategy with greater success and section 6.1.2.3 shows how the utility of key productions changes accordingly. The RMSD is 0.078 considering between participants and the model considering the Move→CF interaction.

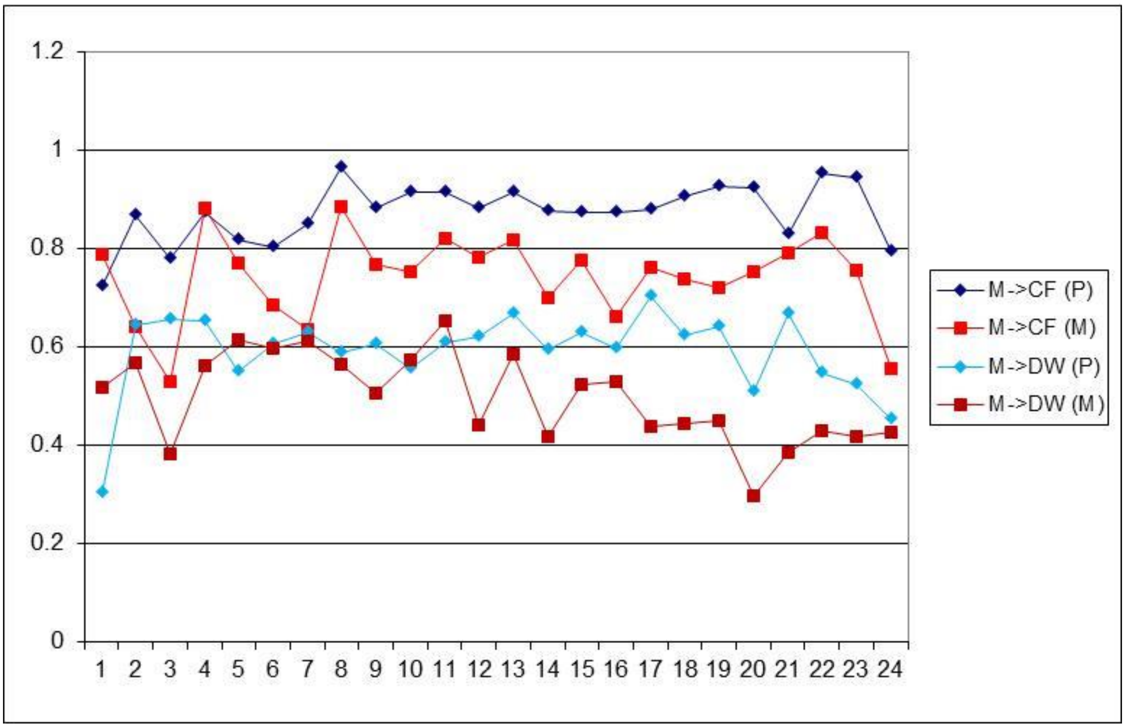



**Figure 5.4: Proportion of time (Y-axis) that participants and the model wait for a unit to finish its movement by trial (X-axis) (M->CF is the sequence of a Move and a Control Fire command, similar case with M->DW)**

Waiting behaviour can be traced to the key productions' utilities shown in figure 5.5. These utilities are extracted from a single model run during the training phase. Waiting for copters to complete a movement is punished in both training programmes. What this model run learned to do is to make short movements with trucks and wait for them until they are ready to execute a CF command. Although this involves wasting some time waiting, this behaviour increases the probability of completing a CF command successfully.

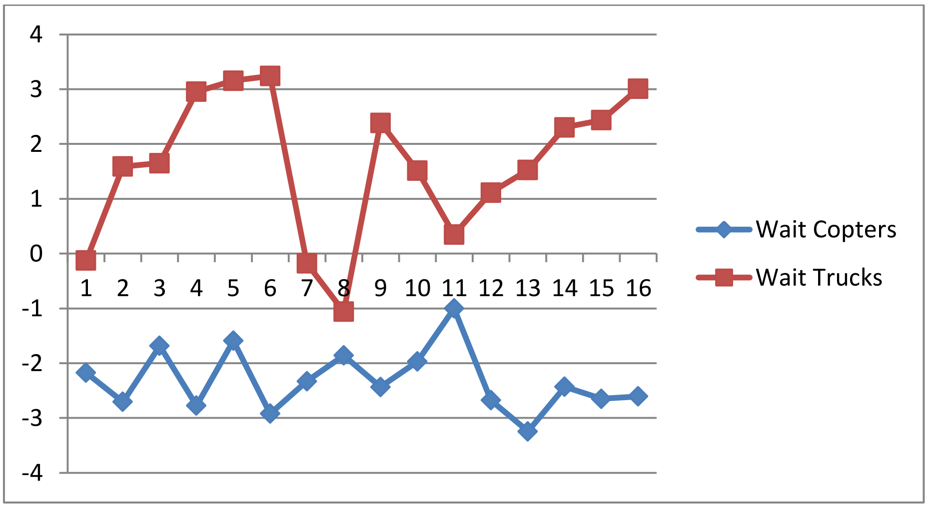



**Figure 5.5: Utility values of productions (Y-axis) associated with waiting behaviour (1 model run) per training trial (X-axis)**

Waiting behaviour is evidence of adaptation to task characteristics. There are no specific rules that prefer waiting over other behaviours: waiting behaviour emerges as a consequence of the reward scheme used in the model. The important fact is that the model is able to pick up this adaptation while controlled by the single-command reward scheme. This result reinforces the idea that focusing on single commands in dynamic tasks represents a valid approach. This behaviour emerges in the model due to the negative rewards received when a CF command cannot be executed, mainly because the fire reaches the location where it was planned to issue it before the command can be executed. The model learns that it is preferable to waste a certain amount of time waiting for the unit to arrive and to issue a CF command as soon as it finishes its movement.

### 5.2.4.4 Achieving good performance

The objective of this section is to provide a description of a good performer from the perspective of the cognitive model. Two factors allow the construction of this description: (1) the fact that this study is competence oriented and hence the model tries to reproduce the behaviour of participants engaged in the task and (2) the way in which the model is designed where micro strategies compete freely and therefore there is no preference for a path of action at the outset. As the model interacts with the FireChief task those patterns of behaviour that contribute to the successful execution of commands stand out by continuously receiving positive feedback. Section 5.3.1 discusses these patterns of behaviour.

| Number | Rule group | Best | Worst | Difference |
|---|---|---|---|---|
| 1 | No switch trucks | -3.13 | -2.86 | -0.27 |
| 2 | Switch trucks | 1.92 | 1.26 | 0.66 |
| 3 | Change unit | -0.49 | 1.29 | -1.77 |
| 4 | No switch copters | -2.33 | -2.12 | -0.20 |
| 5 | Switch copters | -3.05 | -3.05 | 0.00 |
| 6 | Wait copters | -0.07 | -2.65 | 2.58 |
| 7 | Wait trucks | 2.09 | 0.48 | 1.62 |
| 8 | Barrier top-down sc | 3.58 | 1.89 | 1.69 |
| 9 | Barrier top-down ln | 3.26 | 2.77 | 0.49 |
| 10 | Barrier bottom-up | 1.37 | -1.75 | 3.12 |
| 11 | Attack low | -0.55 | -0.31 | -0.24 |
| 12 | Attack high | 1.26 | 0.81 | 0.45 |
| 13 | Attack key | -0.26 | -0.43 | 0.17 |
| 14 | Select Strategy Barrier | 9.34 | 0.94 | 8.41 |
| 15 | Select Strategy NBarrier | 1.95 | -0.91 | 2.86 |
| 16 | Select Strategy Stop | -3.25 | -5.44 | 2.19 |
| 17 | Select Strategy Follow | 0.00 | -4.35 | 4.35 |

**Table 5.11: Comparison of utility values of key productions between the best and the worst performers**

This analysis of strategy adaptivity and attainment of good performance focuses on the dynamic aspects of the process in which productions are tuned. Section 4.2.3 explains the methodology followed in conducting this analysis and the reasons for conducting it. With the aim of gaining insight into what differentiates best and worst performers, two profiles were created based on utility values for each group (model runs of the CT scenario are used in this comparison). To obtain these profiles the utility of relevant productions for each group is queried at the end of the training phase and averaged. The comparison is focused only on the rules relating to the creation of a barrier: the way trucks are used, how they are moved, and how the barrier is created. Utility data from the top and bottom quartiles were compared. Table 5.11 shows the quantitative data related to the upper and lower quartile model runs. This comparison shows that the most striking difference between good and bad performers is that good performers successfully combine top-down (sc = semicircle, ln = line) and bottom-up processes to create a barrier, while the worst performers apply only top-down processes successfully, failing to combine them well with bottom-up processes so that cells selected for the fire-break prove less effective.

Top performers also have a preference for waiting for trucks to Move (they link two commands with the same truck more often) and have a strong preference for using the Barrier strategy. Model-runs in the lowest quartile do not have a preference for waiting for trucks and have a preference for Barrier but not as marked as top model-runs. This example illustrates how cognitive modelling can be used to determine which microstrategy is the most successful by means of utility comparisons.

## 5.3 Summary of QOF

In relation to performance the model captures well the overall performance levels (section 5.2.1), the learning effect produced by the CT (section 5.2.1) and the effects that the different environmental changes generate over performance (section 5.2.1.2). Section 5.2.2.1 demonstrates that performance in the CT group is linked to strategy use and that the model captures this interaction. The model also captures significant interactions related to command use: the faster execution of CF commands when barriers are being created (section 5.2.3.1) and of DW commands when copters are used (section 5.2.3.2). The interaction between performance and strategy use is also captured in the VT condition but mainly for the structured strategies Stop and Barrier, presumably due to differences in the complexity of trials (section 5.2.2.2). An analysis of strategies during the Test phase revealed four interactions related to the phenomenon of cognitive inflexibility that are well captured by the model:

1. The use of the CF command is disrupted as a consequence of wind direction changes in the CTW condition (section 5.2.2.3.1).
2. The execution of the Barrier strategy is not affected in the CTE condition (section 5.2.2.3.2).
3. The Stop strategy is used more frequently and with more success by participants/model in the VTW condition than in the CTW condition (section 5.2.2.3.3).
4. Performance using the Barrier strategy is worst for participants/model in the VTE group (section 5.2.2.3.4).

The interpretation of these four interactions, together with the patterns observed during the training phase, is that both participants and model consolidate the Barrier strategy during the CT programme. When the wind changes direction the participants/model stick to the use of CF commands even when performance is disrupted (a sign of cognitive inflexibility). When the efficiency of appliances is reduced the consolidated strategy Barrier keeps producing good performance. For participants/model in the VT there is more opportunity to practice the Stop strategy. When wind changes direction the Stop strategy is used more frequently as it represents a better choice (as demonstrated in section 5.2.2.3.3). When the efficiency of appliances is reduced participants/model in the VTE group use the Barrier strategy, nevertheless the level of performance is lower than for those that had the opportunity of consolidating this strategy. As mentioned, the model is able to capture well all these interactions. The total number of commands per trial is captured well by the model, which is an indication that the latency of individual actions is similar for both participants and the model.

It is important to stress that the model is more accurate for predicting behaviour associated with the execution of the Barrier strategy. When barriers are being constructed attention is focused on the units that are creating the barrier and collaboration emerges: the model tends to execute short movements (section 5.2.3.3), wait for the units to arrive at its destination (section 5.2.3.4) and use the location of previously executed CF commands as a reference to determine the location of the new CF. This is different to the patterns observed when isolated CF commands are executed: attention is switched to other units when the unit is moving, the unit remains idle before the CF is executed and longer mouse movements are required to execute commands. All these patterns emerge as a result of the competition of procedural rules as described in section 4.1.

# 6. Discussion and conclusions

The object of this study, CPS in dynamic domains, is an interesting and challenging one. As the approach to the study of CPS using microworlds was adopted, it was necessary to understand these tasks. From all the characteristics microworlds have their complexity and dynamic component were deemed as the most relevant. Given than in microworlds the stimulus is no longer under the full control of the experimenter and therefore to duplicate the same set-up for every participant is not possible, it was decided to follow the suggestion of Brehmer & Dörner (1993) of focusing the study on strategies and tactics. It was also decided to create a detailed cognitive model of CPS behaviour in order to understand more about strategy use in complex dynamic situations and several studies involved with cognitive models were reviewed, this research revealed that there are very few cognitive models of highly dynamic tasks. There are several aspects related to the use of strategies in microworlds, particularly dynamic ones, which require further investigation, such as what is the nature of these strategies, how are they affected by task manipulation and how does the problem solver make decisions when coping with the complexity and dynamics of such tasks. FireChief, a dynamic complex task, was selected to conduct the study and the dataset of the Cañas et al. (2005) study was chosen due to its richness and emphasis on strategy use. Because computerized scenarios tend to produce a lot of behavioural data a tool (PAT) was created to facilitate a brand new analysis of the data. Using this tool it was possible to identify different patterns of behaviours which are the components of the various microstrategies modelled in a later stage. The core of this model is its ability to adapt as a response of the feedback it continuously receives from the environment. It was shown how this model was able to replicate several aspects of the empirical data. The aim of this chapter is to go over what has been learned from the FireChief model specifically and, as several aspects of CPS were involved in the creation of the model, to relate these findings to similar results gathered from the literature in order to discuss how this work has advanced our understanding of CPS.

## 6.1. General Discussion

Chapter 1 introduced specific research questions related CPS. These questions are answered in the light of what was discovered during the course of this research.

### 6.1.1. What characterizes strategies in complex dynamic tasks?

A complex dynamic problem involves achieving several goals that are not clearly defined by allocating limited resources to select and execute a variety of actions which can only be implemented under certain constraints and may produce side effects. On top of these features the problem must be solved under time pressure. Section 2.3.2.1 conceptualizes complexity in terms of the number of elements present in the system plus the number of possible actions available to the problem solver. This conception of complexity is shared by several studies using a variety of microworlds (section 3.1). In some microworlds such as FireChief complexity and dynamics are heavily interwoven. Section 2.3.1.5-C explains why complexity and dynamics in FireChief is mainly related to wind conditions. Complexity increases as commands must be issued at a fast pace to fight stronger fires. These strong fires destroy cells quickly and propagate to adjacent cells increasing the number of cells on fire quite quickly. A consequence of this increment in the number of cells on fire is that the problem solver needs to choose a single action from a broader range of them. CPS situations in dynamic domains can be tackled by implementing the right strategies. The characteristics of these strategies are discussed in sections 6.1.1.1 to 6.1.1.4.

#### 6.1.1.1. Focus attention on important aspects of the problem

Several researchers observed that performance in CPS situations decreases as more variables are involved in the task (Hussy & Graznow, 1987; Ackerman, 1992; Schunn & Reder, 2001; Joslyn & Hunt, 1998). The first characteristic of a successful strategy for complex dynamic situations is that it be constructed on the basis of an adequate representation of the problem that abstracts the most relevant aspects of the problem. Of the myriad aspects of a FireChief display, the FireChief model focuses its attention primarily on the fire-fighting units and the fire. The most important piece of information related to fire-fighting units is their status, that is, whether they are available or not for executing commands. Overall the model has about twice the number of rules for perceiving the location and intensity of the fire than for perceiving the location and status of the fire-fighting units, and in general terms more time is spent examining the location of the fire. The location of fire-fighting units is also relevant when considering their proximity to the fire. Problem representation is directly linked with the use of WM (see section 2.3.6.3). The FireChief model holds in WM a chunk that represents its current intention (see section 4.2.3.1) and keeps track of the individual intentions of each unit related to the execution of commands. Keeping the status of units in WM is important otherwise the model must execute a series of perceptual and cognitive actions every time it switches attention to a previously unattended unit in order to determine its status. Other models use a similar approach, for example, in the Table Decision task model (Veksler et al., 2007) the imaginal buffer was used for storing the highest value seen so far.

Because a problem’s representation is centred on the status of individual units, strategies are used basically for making two decisions: whether to issue a DW or a CF command, and where to Move a unit to issue the selected command. Although the focus of perceptual actions is the fire, the results of these perceptions do not become part of the mental representation of the world, but are used to select between firing a variety of different cognitive actions. The model is able to deal with different fire-fighting scenarios by means of continuous perceptual actions and a fire-fighting-unit-centred representation of the world. Using this representation FireChief tasks can be reduced to selecting a general intention and assigning individual commands to each unit. The EnCoRe model (Niessen et al., 1999, see section 2.3.6.1) follows a similar approach and focuses its attention on the aircraft’s vertical position.

#### 6.1.1.2. Use perceptual actions intensively

As mentioned in previous chapters, the rich environment provided by CPS tasks such as microworlds can be used as an External Memory (EM) and it has been observed that the problem solver calculates the cost of accessing this EM and then decides either to use EM or internal memory (Fu & Gray, 2001; Gray et al., 2005; Veksler, Gray & Schoelles, 2007). The second characteristic of successful strategies is their ability to continuously search for elements and make attention shifts to harvest information from the environment. The

FireChief display is demanding for the ACT-R visual module and, because the environment is continuously changing, it is necessary to act fast. Perceptual and motor mechanisms interact with cognitive mechanisms to determine performance. The model needs to distinguish landscape elements, including fire intensity. There are two available model operations for identifying FireChief elements: by physical attribute and by location. The first kind of operation is based either on an item's colour (it works for all FireChief elements) or the alphanumeric labels of the cells (used to check fire intensity). The second kind of test is based on an item's (relative/absolute) location. The FireChief model uses a similar approach to the Argus Prime model (Schoelles & Gray, 2000; section 2.5.2) to find elements based on location: manipulating the scan area for the search. The model needs to make good use of the visual buffer and apply attentional shifts wisely as they provide up-to-date information about the state of the simulation. The model needs to perceive changes in the environment, encode these changes (in the imaginal buffer) and use this information to execute commands or to gather more information. The time required for processing visual elements depends upon the time it takes to switch attention and harvest visual features. The FireChief model also uses a mechanism embedded in ACT-R for detecting unusual developments of fire (subsection 15 in section 4.3.1).

The model confirms the observation of Gonzalez et al. (2004) that performance in FireChief is associated primarily with the ability to store and process visual or spatial information. The ability to process visual information was discussed in the previous paragraph; spatial ability, understood as the ability to process the location of elements in making decisions, is realized in ACT-R by a combination of perceptual and cognitive actions. Spatial stimuli can be encoded by making comparisons between the coordinates of visual elements and can be remembered using the imaginal buffer. The built-in mechanism to locate the nearest element to the current focus of attention is particularly useful for this end. The model also makes use of the aural module to deal with alarms. Effective response to alarms is important for good performance in FireChief. Frequently the model's attention is focused on a unit while another is threatened by the fire, and the only way of noticing this is by processing aural stimuli.

#### 6.1.1.3. Rely on fluid motor actions

Psychomotor ability is a determinant of good performance in a dynamic task, as observed by Rehling et al. (2004). In dynamic tasks any benefit derived from a decision decreases with the amount of time it takes to be executed. Even though an adequate action for a specific situation is chosen, the problem solver must issue a considerable amount of commands and so the execution of this action is quite dependent on manual actions. The third characteristic of successful strategies is its reliance on fluid motor actions. The importance of motor actions is discussed in section 4.2.4, in particular section 4.2.4.1 showed how important the duration of manual actions is for the execution of a Move command. Veksler, Gray, & Schoelles (2007) came to a similar conclusion: in the table cell task 80% of the time between cell clicks was comprised by the motor component. St. Amant, Freed, & Ritter (2005) also found that the motor actions dominated success in a telephone dialling task.

#### 6.1.1.4. Promote adaptiveness

The problem solver must also cope with the dynamic nature of the task. In order to achieve this it is necessary to adapt behaviour to the changing environment. Among others, Bettman (1979) stresses the importance of being receptive to the environment. In addition Schunn & Reder (1998) found that individuals experience different success base rates for each strategy and learn to prefer different strategies. When implementing a strategy by executing commands the FireChief model is primarily driven by the current state of the fire. This information is obtained by perceiving the environment. The model obtains the best performance when there is a combination of the top-down and bottom-up modes of controlling behaviour, suggesting that this approach is beneficial when dealing with dynamic tasks. To adapt to the changing environment the problem solver should be able to monitor its actions and evaluate them. This topic is further discussed in section 6.1.3.

### 6.1.2. How strategy use is affected by task manipulations?

Section 2.2.3.4 explained how task manipulations such as graphical cues, the content of instructions, the cost of accessing information, and primarily the configuration of trials affect strategy use by placing constraints on how actions are executed. What is particularly relevant in this respect is to understand precisely how these manipulations affect strategy use. According to Veksler, Gray, & Schoelles (2007) microstrategies evolve throughout the execution of the task. Also Bettman (1979) argues that strategies develop in an ad hoc fashion during the course of the problem solving process and the means of generating these strategies is by being receptive to the environment. This would appear to be borne out by the way the model tunes production utility values to the different experimental conditions. This research further explored a data set provided by the Cañas et al. (2005) study. The model provides an explanation of how the different training programmes of the Cañas et al. study facilitate or hinder the ability of participants to cope with changes in the environment introduced during the testing phase. This section describes the findings obtained through this study in relation to how task manipulations affect strategy use, which in turn affect task performance. Subsection 6.1.2.1 starts by recapitulating the most relevant aspects of the strategies found during the new analysis. Section 6.1.2.2 describes how task manipulations affect strategy consolidation and its relevance to task performance. Section 6.1.2.3 describes how strategy consolidation affects the way barriers are created and prepares the discussion presented in section 6.1.2.4 about cognitive inflexibility. Section 6.1.2.5 closes this discussion and links the previous topics by relating strategy use to performance.

#### 6.1.2.1. Strategies

Several studies using complex tasks were able to identify strategies used by participants (Charman & Howes, 2002; Veksler, Gray & Schoelles, 2007; Fu & Gray ,2006; Schunn & Reder, 1998, among others). It has also been observed that appropriate strategy selection is key for successful performance in complex tasks (Lee et al, 1995; Byrne & Kirlik, 2005). In order to select an adequate strategy it is necessary to abstract the most relevant aspects of the task (section 6.1.1.1) and, in the context of dynamic tasks, this research suggests that it is particularly important to consider the variable that drives the dynamic component of the task the most. In the context of FireChief wind conditions represent the most important source of information for strategy selection.

The research presented here uncovered a new set of FireChief strategies and helped in the understanding of how they are selected and how their execution is modified as experience is gained. These strategies are non-compensatory (see section 2.2.3.5) as they do not consider all available information before making a decision, a condition that should be expected in this kind of task. In addition, all four strategies are also strong problem solving methods in the sense that they make use of domain-specific knowledge (e.g. how to execute FireChief commands and how trucks and copters operate). Differences in strategy adaptation are a consequence of training programmes, the feedback received after executing actions and trial performance. The model is run with the same parametric values every time and there are no differences in knowledge because all strategies are available to the model at all times. Strategy use is driven by environmental cues and rewards received from the environment in a rather short temporal window. This study also provides a detailed description of how microstrategies are executed by a combination of cognitive, perceptual and motor operations.

#### 6.1.2.2. Strategy Consolidation

The context of CPS shapes the decision making behaviour of the problem solver, this context is a product of the manipulations made to the task. When the task is manipulated in such a way that the problem solver interacts with the same situation over and over there is a particular learning effect. Section 2.2.3.1 describes how skill acquisition is a process that starts in the cognitive stage and ends in the automatic stage. This research explores how this learning process occurs within a highly dynamic scenario.

An approach that interrogates the ACT-R sub-symbolic level (introduced in section 4.2.3) is used to understand more about strategy consolidation as learning in ACT-R is closely related to the tuning of its subsymbolic processes (Anderson et. al, 2004). These utility values of productions are tuned throughout the trials in a unique fashion, under constraints imposed by the properties of the FireChief task, the procedural knowledge represented by rules and rewards from the environment. In ACT-R terms, not knowing what to expect from your actions is equated with balanced utility values in the competing rules for a particular decision; when strategy consolidation occurs, the model 'knows' what to expect from its actions. According to the theory, sufficient practice allows for a change between cognitive stages (Ackerman 1988; Taatgen, 2005). At the level of the cognitive model, a strategy is consolidated when the competition between rules at the different decision points is avoided due to the dominance of a given rule (or set of rules).

More evidence of the consolidation of strategies in the CT condition is the significant increase in performance between the first and last 4 trials (see section 5.2.1). This is an indication of the opportunity the CT trials offer participants to improve performance. This performance improvement can be a consequence of better strategy selection or implementation. The model supports the view that this performance increment is attributable to differences in strategy which are a product of how productions are being rewarded. Lee, Anderson & Matessa (1995) explain performance improvement in an ATC task based on strategy change also: as practice increases participants reduce the number of keystrokes required for completing the task. Similarly, Charman and Howes (2002) observe that practice in the task increases efficiency in the use of commands. In the CT condition, the model executes the most successful strategy with more frequency and with more efficiency as trials pass by. In the CT condition several model runs follow a strategy selection pattern in which the model tries one or two different strategies before executing the Barrier strategy. When the Barrier strategy is executed, acceptable performance is obtained which in turn promotes the selection of the Barrier strategy again, allowing a considerable amount of practice in the creation of a barrier and so forth.

Another phenomenon observed in this research is that consolidation of strategies can be deterred by manipulating the task in such a way that participants do not face the same CPS situation in each trial. From the cognitive modelling perspective this means giving no opportunity for a particular rule or rule-set to become dominant due to changing environmental conditions that make the outcome of actions less stable. The two training programmes in the Cañas et al. (2005) study have different patterns of complexity variation where the hardest trials lead to more strategy exploration. For this reason the VT programme promotes a fairer competition between strategies because more options are explored in comparison to the CT condition where it is hard to obtain good performance other than for the Barrier strategy, thus favouring the choice of this strategy over others. In the VT condition there are more complexity fluctuations, more changes, and a less predictable reward pattern.

When the training phase is over what distinguishes participants from the two training groups is the different utility values acquired for the same set of rules. These different utility values represent not only strategy preferences but also a preference for how to implement those strategies. In this sense the knowledge that the model acquires through its interaction with the FireChief simulation is stored as a set of utility values. These different utility values generate different responses to the changes in the environment that are introduced during the testing phase. The problem solver has to resolve the tension between exploration and exploitation throughout the problem solving activity (Nellen & Lovett, 2004). Using the terminology of Nellen & Lovett, in the VT condition the model cannot establish sufficient trust in an option in order to exploit it.

#### 6.1.2.3. The Barrier case

The previous section explained how strategy consolidation may occur if certain conditions are present, this section describes in detail strategy consolidation related to the creation of a barrier of CF commands and prepares the discussion of cognitive inflexibility in the following section. The most structured strategy is Barrier and this also presents a richer set of elements for cognitive modelling because the three FireChief commands are frequently used. The question of whether it is a good option to deploy a barrier depends on many factors, the most important being wind conditions. If the wind strength is of 6 or higher creating a barrier is a good option. The analysis of production rule utilities provides an explanation of how the different training programmes give rise to different ways of creating the barrier: the particular method of creating a barrier depends on previous ways of constructing the barrier within several contexts and thus on the different weights accrued by production rules associated with the creation of the barrier. Contrastingly, if wind strength is too low (3 or less) it is more advisable to attack the fire with DW commands and barrier construction rules are not rewarded.

During CT participants tend to use and perfect the Barrier strategy. The participants and model respond to the CT programme by learning to create a barrier to stop the fire, and successful participants in the CT group are those that specialize in the creation of barriers. When a barrier is being created, the mouse pointer tends to be closer to the next target cell and, because a barrier is usually constructed by a sequence of CF commands, the distance of the various mouse pointer movements tends to be shorter in comparison to the NonBarrier strategy where CF commands are more dispersed. As a result, although the model burns extra cognitive resources in identifying the next cell of the barrier (compare subsection 5 in section 4.3.1 and subsection 4B in section 4.3.2), the longer times associated with moving the mouse pointer when the NonBarrier strategy is executed causes higher latencies. Table 5.5 showed that a CF belonging to a barrier takes longer to execute in the VT condition. The explanation offered by the cognitive model is that extra cognitive and perceptual steps are being made before issuing the CF command: the VT group pays more attention to the developing fire in comparison to participants in the CT group and therefore exhibits more reactive behaviour. Overall the VT predisposes responsive behaviour.

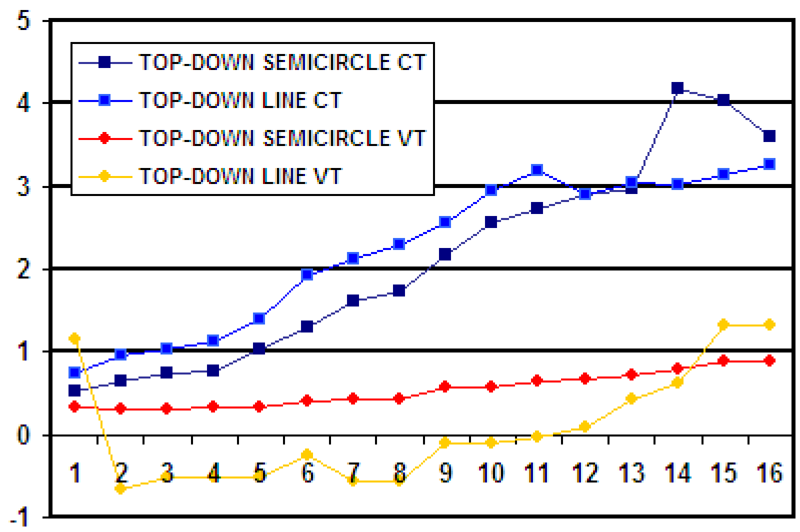



**Figure 6.1: production utility values of barrier creation (top-down) related rules**

Figure 6.1 shows how the utility of the set of productions related to the creation (in a top-down fashion) of the barrier increases as trials are completed for the CT condition. The utility of these productions for the VT condition is also shown. The set of production rules available for use are exactly the same for both training programmes, (that is, a single model undergoes either of the training conditions). However, the pattern of utility change is different as can be seen in the graph. The continuous increase of production utility values for rules associated with the Barrier strategy means that their repeated choice (related to the execution of commands) is continuously rewarded, a phenomenon that only occurs in an environment that does not dramatically alter the effect of the actions being executed on repeated trials. In the VT condition, the same chain of actions that successfully executes a CF command in the CT condition may fail, for example, when wind strength is very high. The invariable characteristics of the CT condition allows the discovery of the production rules that work best for the CT trial and once these rules are discovered they are selected over and over again. In the VT condition, the firing of the same rules may not receive the same feedback.

According to Veksler, Gray & Schoelles (2007) time comparisons between commands provide insights into how strategies evolve. The model ends up preferring a top-down approach to the creation of the barrier in the CT condition. This switch from the competition between bottom-up and the top-down approaches towards the dominance of the top-down approach shaves-off time as a consequence of the reduction in perceptual actions. In dynamic tasks the problem solver needs to control how much information will be gathered before executing a command. Fu & Gray (2006) found that people decrease the number of information-seeking actions when their costs are increased. Although the Cañas et al. (2005) experimental design does not manipulate the cost of information-seeking actions there is a cost associated with performing visual searches. Similarly to the case of Fu & Gray (2006) where utility of information was manipulated, the utility of gathering information related to the location of fires is indirectly manipulated by the training programmes. In this sense the utility of gathering this information (fire location) is low when no change in wind strength is expected, which is the case of the CT. The model is able to reduce the amount of information that needs to be gathered in order to make a decision when environmental conditions are constant (such as in the CT condition) in comparison to a more variable situation (such as the VT condition) regardless of the amount of practice. In other words, the model is sensitive to the level of dynamics of the task and is able to adapt the amount of perceptual actions to environmental conditions.

#### 6.1.2.4. Cognitive inflexibility

Schunn & Reder (2001) consider the possibility that once a strategy is consolidated it becomes less adaptive in response to success and failure feedback. Direct evidence for this was found: as a consequence of the CT condition, the model shapes the utility of its rules in such a way that it becomes insensitive to rewards at the micro-level for some time. For this reason larger changes at the global level are required for generating the required strategy change during the test phase. Strategy selection during the test phase depends on various factors: the type of training, strategy preference before the change, and type of environmental change. Particular strategies confer particular benefits in the different training and testing phases. Barrier is most effective in the CT condition, no strategy is preferred in the VT condition. Similarly, Stop is most effective in the *Wind Change* test condition, whereas Barrier is most effective in the *Efficiency Reduction* condition (see section 4.3.4). The model running under the CT is slower to change strategy when the wind changes direction. A feature of the data is that participants from the CTE group perform better using the Barrier strategy than participants in the VTE group (see table 5.3). Participants (and the model) in the CT condition can be cognitively inflexible to their advantage in the efficiency reduction condition. Remember that participants from the CTE group have more opportunity to practice the most successful strategy while the higher strategy variation of participants in the VTE group deters the tuning of the Barrier rules to the structure of the environment. In the testing phase a change in wind direction can be successfully addressed by any structured strategy while a change in the efficiency of appliances requires the use of the Barrier strategy. This can be concluded by observing that the model is able to obtain good performance in the WD condition by either executing the Barrier strategy or the Stop strategy. In the WD condition, participants in the CT group (CTW) still use the Barrier strategy quite frequently, while participants in the VT group (VTW) use the Barrier strategy less frequently. The pattern of strategy use when efficiency is reduced is very similar between the CTE and VTE groups: Barrier is the most preferred strategy and both groups repeatedly use it.

Cañas et al. (2005) found that the changes introduced in the environment during the test phase generated significant effects in performance, but also that these changes affected the participants differentially depending on the strategy that they were putting into practice. Cañas et al. hypothesized that strategies that rely on the execution of CF commands are more affected by changes in the direction of the wind, while strategies that rely on the execution of DW commands are more affected by changes in the efficiency of appliances.

The detailed specification of strategies (Section 3.3) implemented in the model provides the fine grain level of detail necessary to understand precisely how these changes impacted performance. In the situation where the wind changes direction the model needs to fire a greater number of rules in order to identify and select the next cell in which to issue a CF command thus increasing the number of problem solving steps. What happens is that after a candidate cell is chosen there are rules that check the location of fires nearby. When wind changes direction these rules end up encouraging the model to choose a different cell based on the location of the fire. The result is that the model spends more time deciding how to create the barrier. We can use this detailed understanding of model behaviour to understand the differential impact of a change in wind direction for the model undergoing CTW compared to the model undergoing VTW (where the VT model has developed a preference for the Barrier strategy, so that it is directly comparable to the CT model). The model trained in the CT condition has a clear preference for the use of top-down approaches while the model trained in the VT condition has a preference for the bottom-up approach. This difference has an impact when the wind changes direction at second 60. In the VTW condition the model is continuously observing the fire in order to identify the next cell in which the CF will be issued, so when the change in the wind occurs, the model selects the next target cell based on more accurate information. On the other hand, a model that prefers the top-down approach to the creation of the barrier will place the next block of the cell without recourse to observing the fire. In this sense the automation of the strategy runs the risk of deterring the problem solver from extracting relevant information from the environment, and hence allows the emergence of cognitive inflexibility. We can compare this with the situation in which appliance efficiency is diminished and therefore attacking large fires with water is no longer feasible and therefore creating barriers using DW commands is no longer viable. In the EF condition there are several fires with intensity 3 or higher (outside the capacity of the DW command); if CF commands are not issued, performance will be inevitably low. Although the model may start using an Only-DW strategy such as Stop it will receive multiple alarms (negative rewards) when trying to extinguish strong fires and the consequence will be that a new approach to solve the problem will be adopted.

#### 6.1.2.5. Strategy use and performance

To conceptualize strategy use as a product of previously learned utility paves the way to understanding more about performance differences. Section 5.2.2 discusses task performance both from participants and the model. The best and worst performers reflect a different pattern of utility values in the productions used for the creation of a fire barrier. The model reveals that problem solvers facing a task under time pressure will be more successful if they combine top-down and bottom-up processes to deal with the task. The best FireChief performers have a preference for waiting for trucks to complete a move then immediately issuing a command (that is, they link two commands with the same truck more often) and have a strong preference for using the Barrier strategy. Less strategy change is related to higher performance in the CT programme. To avoid wasting time is a good practice, the best performers do not waste a lot of time moving the mouse long distances. Lee et al. (1995) observe that the strategy use of participants contributes significantly to performance. In general terms the best performers are those that choose the best strategy frequently. Frequent use of a strategy enables its refinement through the learning of convenient behaviours such as waiting for trucks to complete a move if the length of the movement is short, the most appropriate form for a barrier, and the appropriate distance between a barrier and the fire.

The initial strategy selection has an effect on performance. Consider a model starting the CT programme. If the model selects the Barrier strategy it is possible that its performance will be high enough to increase the probability of selecting the Barrier strategy a second time round (which may result in further good performance). On the other hand, if the model selects the Follow strategy it is probable that its performance will be low increasing the probability of selecting another strategy in the following trial. The important point is that the model is able to explore new strategic alternatives if it receives a low final feedback, that it tends to repeat a strategy selection that has proven successful in previous trials (taking into consideration the assessment of the situation) and that it tends to improve the implementation of a strategy the more it is practised. Payne et al. (1988) argue that a major problem is to understand and predict when a particular strategy will be used.

### 6.1.3. How do choices arise in complex and dynamic situations?

Ultimately, models of human behaviour involving decision making look for explanations of how these decisions are made. Section 4.1 described two cognitive modelling paradigms: Competing Strategies and Perceptual and Motor Processes. Due to the necessity of implementing these paradigms in the context of a highly dynamic task this study enriched them. Following the Competing Strategies paradigm four strategies compete to solve every FireChief trial and the relative merits of the different strategies are managed by the utility learning mechanism. Following the Perceptual and Motor Processes paradigm the model heavily relies on perceptual actions in order to maintain an updated representation of the task environment. Nevertheless, there are some characteristics of the FireChief task, mainly related to its dynamic component, which posed specific modelling challenges and hence demanded a novel approach.

In the context of the cognitive model top-down control refers to the definition of strategies plus the sequences of productions that implement chains of behaviour. A strategy is specified at a fairly abstract level. In the case of the Barrier strategy the plan is to start using trucks to create a barrier and at the same time to use copters to extinguish some of the fires; after the barrier is completed all units start attacking the fire. On the other hand bottom-up control refers to the processing of feedback that tunes the utility value of each production plus the deliberate act of sensing the state of the world for gathering information.

This research found that the way environmental feedback is processed is critical for controlling behaviour. The model continuously gathers information and re-shapes the execution of its strategies. Selecting actions based on utility comparisons facilitates a fluid and quick selection of actions which is instrumental for obtaining good performance, particularly in situations with high levels of dynamics and time pressure. This ability to adapt depends on how the utility value of productions are modified which in turn is based on the model's design, where several rules compete in selecting the next action at almost every time step, and the rewarding scheme.

Participants interacting with the FireChief task show a considerable diversity of behaviours. A modelling challenge was to allow this richness to appear in the model behaviour and this was found to be possible by a promoting the competition of microstrategies at the sub-strategic level. This characteristic of the model distinguishes it from models such as that, for example, presented by Taatgen (2005) where actions follow a pre-established plan. The cognitive model has knowledge of how to execute atomic actions that can be combined to issue commands, but the precise sequencing of these actions is largely left to the reward history of individual rules. The dynamic nature of the FireChief task has a considerable influence in this matter: the ever changing environment in FireChief favours the reaction to environmental cues over the creation and execution of detailed plans, at almost all times the model must choose a single action from a pool of various options (check the example in section 5.4.2). The task of achieving the model's flexibility in a dynamic task such as FireChief represented a complex task mainly due to the potentially high degree of brittleness. It has been found that brittle models fall short in accounting for flexibility (Taatgen, 2005) and the dynamic nature of FireChief produces a large variety of situations in which the model needs to know what to do. The model has to handle all possible events triggered by FireChief whilst enabling the execution of commands. The occurrence of distracters made this task more complex. The model is able to recover from distracters by relying in its working memory (cf. Brumback et al., 2005) which is stored in the imaginal buffer. Section 6.1.3.1 discusses the core of the extension implemented to the *Competing Strategies* modelling paradigm to enable flexibility whilst enabling the completion of high-level goals.

#### 6.1.3.1. Rewarding the execution of commands

When a command is successfully executed it has an impact in the environment. This research highlights the key role of the size and distribution across rules of rewards. As pointed out by Janssen, Gray & Schoelles (2008) the conceptualisation of reward used during the development of the model greatly influences its behaviour. In the FireChief model microstrategies are selected by applying rational analysis: those rules that contribute the most towards the successful execution of commands are rewarded. By rewarding the successful execution of commands, the FireChief model emphasizes the relation of overt environmental changes (those produced by the execution of commands) and rewards. The model is not driven by the completion of large tasks or elaborated high-level goals (section 4.2.3.1), but rather the model shows that it is possible to gain control over complex dynamic situations by focusing on the execution of atomic actions whilst following a loose strategy definition.

The reward scheme works along with the random component in ACT-R to achieve this flexibility. The use of the *s* parameter (noise) is important to show variability, as in the Peebles & Bothell (2004) model (section 2.5.2). As pointed out by Jones, Ritter & Wood (2000) increasing noise during the calculation of utilities reduces the influence of knowledge about which strategy is the most effective. In their model of the Tower task they increase the noise parameter systematically and found that a value of 6 (which is higher than the standard value of 3 reported is several studies) gave the best fit to data. With a noise level of 6 any model is likely to select incorrect strategies, this is similar to what happens during VT where the lack of opportunity for rehearsing a single successful strategy allows for a freer competition between strategies rather than the dominion of a single strategy observed in the CT condition.

Rewards can be seen as an abstraction of the ability of an individual to process feedback from the environment. The objective of the Tower task (Jones, Ritter & Wood, 2000) is to build a pyramid using 21 wooden blocks (based on a target pyramid presented to participants at the beginning of the trial). If a rule helps to generate the construction the model believes to be correct, the rule's strength is increased. A similar approach (that is, to reward certain actions) is followed by the FireChief model. An important difference between the Tower task and FireChief is the degree of time pressure. In FireChief decisions must be made at a fast pace because the environment is continuously changing, so there is not the same opportunity for evaluating options as in the case of the Tower task. The consequence is that, even when the right decisions are made, rewards are diminished if they are not executed in a timely manner.

An interesting aspect of modelling a task that spans several minutes is how different time bands of behaviour are impacted. Considering Newell's (Newell, 1990) time scales, the FireChief task covers both the rational and the cognitive bands. The model performs actions at the cognitive band level: deliberate acts (100 msec.), operations (1 sec), and unit tasks (10 sec). A unit task refers to the execution of single commands. The model also performs actions at the rational band: each FireChief trial lasts 4 minutes and each participant completes 24 trials (96 minutes in total). Within this time it is necessary to integrate sets of unit tasks into blocks of behaviour, such as creating a barrier, both in order to complete a full trial and to complete entire training and test programmes. The cognitive model shows how decisions at the cognitive band level impact the rational band. Using command execution reward as an example, waiting behaviour in the *continue barrier* microstrategy impacts the execution of the Barrier strategy: waiting for trucks to travel short distances (cognitive band) increases the probability of issuing a successful CF command (cognitive band) which in turn increases the probability of implementing the Barrier strategy (rational band) and complete a full training programme (rational band).

## 6.2. Future lines

This section describes five avenues of exploration that can be followed by reusing the artefacts already provided by this research or by making extensions to them.

### 6.2.1. Modelling poor performers

Although it might be expected that a game-like simulation such as FireChief would maintain the interest of participants, it is possible to identify in the data set problem solvers who almost always present low performance, possibly signalling a lack of effort or interest. Although this research does not attempt to model personality factors, there are moments when there is no rational basis for making a choice and for which personal preferences may influence strategy selection: for example, strategy selection for the very first trial, prior to any performance feedback being received.

### 6.2.2. Running more experiments

The model could be further tested by obtaining more empirical data using new FireChief scenarios and training programmes. To research how much training is required to become cognitively inflexible and to assess the capability of the model to replicate this phenomenon the amount of practice can be varied. The current FireChief model shows cognitive inflexibility after being trained 16 times with the same scenario. A training programme that starts with 12 VT scenarios and finishes with 4 CT scenarios can be used to test (1) whether participants show signs of cognitive inflexibility or not after a shorter training period and (2) whether the model is able to capture this tendency. It may be the case that participants become inflexible after the last 4 trials but the model does not consolidate its strategies in such a short period. This research found that participants tend to use CF commands when wind strength is high and that, with enough practice, patterns that resemble a barrier of CFs emerge (and the model replicates this phenomenon). But, what would happen if the capability to execute CF commands is reduced? Similarly to the EF test condition CF command execution can be affected in such a way that Only-DW strategies (Stop and Follow) should be preferred even though wind strength is high. In this scenario the capability of DW commands to extinguish fires should be increased to allow good performance (otherwise participants will get low performance almost always). In this scenario it can be tested (1) whether participants and the model stop executing CF commands and (2) how long it takes for participants and the model to stop executing CF commands. At the same time it would be interesting to see the level of performance of the Stop strategy after the capability of appliances to extinguish fire increases.

### 6.2.3. Exploring different task manipulations

Several researchers have found that subtle changes in the interface may represent significant cognitive changes for participants (section 2.2.3.4). For instance, this research found that participants, and the model, have a preference for using copters when dropping water on fires. This preference may be caused by (1) the fact that copters are faster that trucks, (2) that copters have slightly more power to extinguish fires or (3) that trucks can be destroyed by the fire. These features of the task can be manipulated to determine whether participants start using trucks with more frequency and to test if the model is able to replicate this behaviour.

### 6.2.4. Incorporating different kinds of data

Studies such as Schoelles & Gray (2000) include the use of eye tracking data, this information helped them to discover a particular strategy in which participants move the mouse pointer over an aircraft, switch their attention to a specific area of the visual display, click the mouse and finally notice a change in the visual display (as a result participants can be sure that the information displayed corresponds to the selected aircraft). Without eye tracking data the discovery of this strategy would have been impossible. The use of eye tracking data may contribute to understanding better how participants deploy their attention and hence the QOF of the model may be improved, including the latency of commands. It would be particularly interesting to know whether eye tracking data can contribute to improving the QOF of the subtask of executing a CF command (just after the unit has arrived at the target location). It would also be interesting to study how the appearance of spot fires and fire development captures attention and compare this new data with the current mechanisms used by the model.

### 6.2.5. Adding more learning mechanisms

Section 4.2.3.2 explains why the production compilation mechanism was not used in FireChief. Nevertheless it would be interesting to explore this path. Also, although utility is able to capture many useful aspects of how choices are made, other constructs may enrich the quality of the model. For instance, Nellen & Lovett (2004) propose that considering the amount of information gain (understood by these authors as a measure of how much knowledge is gained as a consequence of selecting a particular action) is an important factor when making decisions. It would be useful to incorporate a measure of information gain into the model, mainly due to the fact that strategy exploration is important for guiding behaviour. By adding this element the model's focus won't just be the execution of single commands but also the discovery of new knowledge.

## 6.3. Conclusions

The contributions of this work towards our understanding of CPS are the methodological approach to the creation of the model, the design patterns embedded in the model (which are a reflection of the cognitive demands imposed by the nature of the task) and mainly an explanation of how skill, described in terms of strategy use, is acquired in complex scenarios. This study contributes to our understanding about strategy use in complex dynamic tasks: which strategies are used, how they are selected, and how strategy execution changes as experience is gained. A key finding is that good performance is linked to an effective combination of strategic control with attention to changing task demands, reflecting time and care taken in informing and effecting action.

Several artefacts were produced by this effort including a dynamic task fully compatible with ACT-R, a tool for analysing both participant and model generated data, and a cognitive model whose features enable the replication of several aspects of the empirical data. In this research four strategies were identified, and their structure comprising microstrategies clearly described, in a way that allows modifications or improvements to be made in order to measure their impact for the overall strategy. These strategies can be modified to test different approaches to fighting the fire, alternatively different sizes and locations of rewards can be implemented. The model can run under the same task configuration using the same knowledge (rules) and yet produce different results: variability is achieved by allowing competition between rules based on utility.

### 6.3.1. Understanding CPS behaviour from a cognitive modelling perspective

This study provides a deeper understanding of the phenomena observed in the Cañas et al. (2005) study, including a computational realisation of the cognitive inflexibility phenomenon, the focal topic in the Cañas et al (2005) study. This understanding is a product of a deeper scrutiny and new analysis of the data plus the development of the cognitive model. The model describes how ACT-R actions are combined into microstrategies and how these microstrategies are combined to form strategies. Chapter 5 demonstrates that the model captures well the overall performance levels (section 5.2.1), the learning effect produced by the CT (section 5.2.1) and the effects that the different environmental changes generate over performance (section 5.2.1.2). The interaction between performance and strategy use is also captured in the VT condition but mainly for the structured strategies Stop and Barrier, presumably due to differences in the complexity of trials (section 5.2.2.2).

Strategy consolidation and cognitive inflexibility can be traced to the utility values of productions: the different training programmes produce a different profile of utility values. A particular implementation of a strategy depends on the fine tuning of ACT-R rules, as a consequence of environmental rewards, and thus is a product of both the specification of trials and the history of the interactions between problem solver and task. Cognitive inflexibility occurs in the CT condition because the learning mechanism shapes the utility of rules in such a way that the model becomes insensitive to negative rewards at the micro-level for a specific period of time during the test phase. These dominant rules tend to be those that belong to the Barrier strategy for the CT condition. In this sense the automation of the strategy runs the risk of deterring the problem solver from extracting relevant information from the environment, hence allowing the emergence of cognitive inflexibility. In the VT condition no single strategy dominates. Section 5.3 describes four interactions related to the phenomenon of cognitive inflexibility that are well captured by the model during the test phase.

A detailed model of performance allows a better understanding of the role of mouse movements in understanding variability in command duration. In the case of FireChief the spatial distribution of commands determines the time required to move the mouse pointer between cells. At the same time the spatial distribution of commands is a consequence of the strategy that is implemented and the characteristics of the trial. The time required for moving the mouse pointer makes an important contribution to the total duration of commands. This phenomenon is not surprising; nevertheless, the detailed dimensions and characteristics of the impact of mouse movements on command execution was not clear based solely on the analysis of participant data; the model delivered key insights into this phenomenon. As a consequence of accurately modelling these components of command execution several patterns of command use observed in the human data are well captured by the model.

- Longer latencies associated with executing a CF command that is not part of a barrier. In this respect it is important to track down the moment at which the cognitive effort is made (see section 4.3.4). Cognitive effort for placing the new section of a barrier needs to be traced to the time before the movement is executed and not only the point at which the CF command is issued (section 5.2.3.4). Similarly the model captures a difference in latencies between executing a CF command when using the Barrier strategy and executing a CF command when using the NonBarrier strategy (section 5.2.3.1). In this respect the model highlights the weight placed on the time associated with mouse movements in the total time required to execute commands.

- Preference for shorter movements while fighting the fire, particularly when strategies are consolidated (section 5.2.3.3).

- Longer time associated with the execution of the first CF command of a fire-break barrier in comparison with the remaining CF commands (see section 5.2.3.1); sub-section 1 in section 5.3.1 describes what the model does with this extra time.

Other relevant interaction is that, in the VT programme, the model shows a preference for creating a fire-break barrier in a reactive way. Considering the high variability of the VT programme, bottom-up control is the best approach: the variability of wind strength makes it harder to apply top-down control. On the other hand, the model trained in the CT programme chooses to exert top-down control in creating the fire-break barrier. Again, this preference is not a product of symbolic reasoning, but an emergent property of the reward scheme. It was also observed that participants wait for a truck to finish moving in situations where there isn't long to wait so that a CF command can be issued immediately upon arrival. This interaction is also mediated by the type of command issued after the movement: waiting behaviour is typically favoured prior to the execution of CF commands, but not DW commands. This behaviour arises from the trade-off between the advantages of waiting over getting on with fire-fighting elsewhere; there are no specific rules that prefer waiting over other behaviours: waiting behaviour emerges as a consequence of the reward scheme used in the model. This evidence not only supports the appropriateness of using the Competing Strategies paradigm to model CPS behaviour in dynamic tasks, but also of extending this paradigm with an additional layer as described below.

### 6.3.2. Extending the Competing Strategies cognitive modelling paradigm

This research supports the view that, for dynamic tasks, the competition of strategies is not limited to strategy selection, but also to the execution of the strategy. The dynamic nature of these tasks forces a second layer of competition between alternative courses of action at a lower functional level (i.e. microstrategies). These processes represent two different kinds of decisions in complex dynamic tasks that should be addressed differently when creating models of human performance. After a strategy is selected the implementation of it requires several quick, non-deliberative decisions within similar setups. This is the recurrent choice problem that the model faces throughout a trial. This finding enriches the *Competing Strategies* cognitive modelling paradigm (section 4.1) by incorporating an additional layer of within-strategy execution competition. This layer is comprised by several architectural features.

The considerable dynamic component of the computer simulations such as microworlds and the large amount of visual elements available in their interfaces pointed towards the exploitation of the Reinforcement Learning mechanism embedded in ACT-R over a declarative approach. The consequence is that cognitive models of complex dynamic tasks become primarily stimulus-driven where retrievals from declarative memory are infrequently required. In such scenarios the model can make use of the task environment as an external memory. This research suggests that, in line with Fu & Anderson (2008), the high cognitive load of FireChief hampers explicit memory encoding but not the Reinforcement Learning mechanism. It can be argued that the CT programme enables the creation of declarative rules to govern behaviour, but this would mean a different set of rules for the different models. Nevertheless the existence of such rules can also explain cognitive inflexibility as their top-down influence would generate insensitivity to feedback. More research is needed in this respect. In the end the model's behaviour is driven by external environmental cues but mediated by internal cues represented as production utility values.

Fu & Anderson (2006) stress the importance of identifying the critical choices responsible for the delayed reinforcement, in this context one of the greatest challenges was to find the right level of abstraction for updating the utility of productions. Fu & Anderson (2008) consider that dynamic tasks increase the complexity of the credit-assignment problem. Reinforcement Learning is driven by the gradual accumulation of experiences through trial-and-error feedback to inform the correctness of future choices.

Rewarding productions for their effectiveness in successfully completing commands has proven to be a good criterion (section 4.2.3.1). This scheme enables the differentiation and tuning of key rules by providing the right level of granularity. This reward scheme presents emergent properties in the model's behaviour and performance that mirror behaviours observed in the human data. These behaviours are not pre-programmed by means of productions rules but result from the way rules compete and are rewarded. The model of the Blocks World Task described in Janssen & Gray (2012) shows adaptation to its accuracy goal by keeping the number of placed blocks low. Similarly the FireChief model adapted to its goal by striving for the successful execution of commands that ensures the continuous use of available resources. These results shed some light onto how people deal with complex, dynamic tasks. If a particular way of implementing a microstrategy is continuously rewarded, it becomes the preferred way of executing it, regardless of the possibility that another action might be more adequate. This observation can be extended to other dynamic tasks where the time for planning is severely reduced due to time pressure. For instance, instead of explicitly selecting which microstrategy to execute for each subtask, the Argus Prime model (section 2.5.2) may reward the execution of single commands, such as clicking on an aircraft, and allow a free competition of all microstrategies.

Results also suggest that a form of local optimization (the execution of single commands) result in global optimization (stopping the fire) under certain constraints: selecting the appropriate strategy and exerting a weak amount of control over strategy execution. The weak control structure of the strategy definition is similar to the one described in Taatgen (2005) about an Air Traffic Control task. But rewards, ultimately, only change the utility of productions: it is also necessary to provide a suitable design to allow the emergence of adaptive behaviour. The enforcement of production competition embedded in the basic workflow (figure 4.1) ensures that the rules responsible for the decisions are properly credited and, as a consequence, small fluctuations in production utility caused by the reward scheme add up to define a pattern of behaviour at a higher level (e.g. the emergence of cognitive inflexibility in the CT).

Solving problems is a pervasive aspect of everyone’s life and a subset of these problems are infused with a dynamic component. The results of this research pinpoint a set of best practices that successful strategies for complex dynamic tasks should have: identify a set of relevant elements of the task and focus attention on them, perceive the status of these elements continuously, execute motor commands promptly, allow flexibility when executing the strategy and look for the execution of those actions with the highest impact on the environment. As a corollary this research supports the idea that, from the multitude of cognitive demands that the problem solver must meet in order to solve complex dynamic tasks, the primary one is the ability to process feedback from the environment. Finally, based on the insights generated by a cognitive modelling approach, this research also proposes that, while dealing with a complex dynamic task under time pressure, the problem solver is focused on completing clearly defined subtasks (such as executing single commands).

## Bibliography


Ackerman, P. L. (1988). Determinants of individual differences during skill acquisition: Cognitive abilities and information processing. *Journal of Experimental Psychology: General*, 117, 288-318.

Ackerman, P. L. (1992). Predicting individual differences in complex skill acquisition: Dynamics of ability determinants. *Journal of Applied Psychology*, 77, 598-614.

Ackerman, P. L., & Kanfer, R. (1993). Integrating laboratory and field study for improving selection: Development of a battery for predicting air traffic controller success. *Journal of Applied Psychology*, 78, 413-432.

Ackerman, P. L., & Cianciolo, A. T. (2002). Ability and Task Constraint Determinants of Complex Task Performance. *Journal of Experimental Psychology: Applied* 2002, Vol. 8, No. 3, 194–208.

Alberts, D., & Hayes, R. (2002) Understanding command and control. *CCRP Publication Series*.

Alibali, M., & Booth, T. (2002) Representation Strength Influences Strategy Use and Strategy Discovery. *In Proceedings of the twenty-fourth annual Conference of the Cognitive Science Society*.

Anderson, J. R. (1983). *The Architecture of Cognition. Cambridge*, MA: Harvard University Press.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah: Erlbaum.

Anderson, J. R. (2000). *Cognitive psychology and its implications*. 5th ed. New York, N.Y. : Worth.

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science*, 26, 85-112.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. (1995). Production system models of complex cognition. *In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 9-12). Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . (2004). An integrated theory of the mind. *Psychological Review 111,* (4). 1036-1060.

Atkin, M. S., Westbrook, D. L., & Cohen, P. R. (1999). Capture the Flag: Military simulation meets computer games. *In Proceedings of AAAI Spring Symposium Series on AI and Computer Games*, 1-5.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61:183-193.

Ball, J. T., Gluck, K. A., Krusmark, M. A, & Rodgers, S. M. (2003). Comparing three variants of a computational process model of basic aircraft maneuvering. In *Proceedings of the 12th Conference on Behavior Representation in Modeling and Simulation,* 87-89. Orlando, FL: Institute for Simulation & Training.

Beckmann, J. F., & Guthke, J. (1995). Complex problem solving, intelligence, and learning ability. In P. A. Frensch & J. Funke (Eds.), *Complex problem solving: The European Perspective* (pp. 177-200). Hillsdale, NJ: Lawrence Erlbaum Associates.

Berlekamp, E.R., Conway J.H. & Guy, R.K. (2001). *Winning Ways for Your Mathematical Plays*, v1, A K Peters.

Berry, D. C., & Broadbent, D. E. (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology*, 79, 251-272.

Bettman, J. R. (1979). *An Information Processing Theory of Consumer Choice*, Reading MA: Addison-Wesley.

Bothell, D. (2007). *ACT-R 6.0 Reference Manual*.

Brehmer, B. (1987). Development of mental models for decision in technological systems. In J. Rasmussen, K. Duncan & J. Leplat (Eds.) *New technology and human error* (pp. 111-120). Chichester, England: Wiley.

Brehmer, B. (1992). Dynamic decision making: Human control of complex systems. *Acta Psychologica*, 81, 211–241.

Brehmer, B. (1995). Feedback delays in complex dynamic decision tasks. In P. Frensch, & J. Funke (Eds.), *Complex problem solving: The European perspective*. Hillsdale, NJ: Erlbaum.

Brehmer, B., & Allard, R. (1991). Real-time dynamic decision making: Effects of task complexity and feedback delays. In J. Rasmussen, B. Brehmer, and J. Leplat (Eds.) *Distributed decision making; Cognitive models for cooperative work*. Chichester: Wiley.

Brehmer, B., & Dörner, D. B. (1993) Experiments With Computer-Simulated Microworlds: Escaping Both the Narrow Straits of the Laboratory and the Deep Blue Sea of the Field Study. *Computers in Human Behavior*; Vol. 9, pp. 171-184.

Broadbent, D. E. (1993). A word before leaving. *In attention and Performance XIV, Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, ed. D.E. Meyer,and S. Kornblum, 864-879, Cambridge; MIT Press.

Brumback, C. R., Low, K. A., Gratton, G., & Fabiani, M. (2005). Putting things into perspective: Individual differences in working-memory span and the integration of information. *Experimental Psychology*, 5291), 21-30.

Buchner, A. (1995). Basic Topics and Approaches to the Study of Complex Problem Solving. In Frensch P.A, & Funke J. (Eds.) (1995) *Complex problem solving: The European Perspective* (pp. 201-218). Hillsdale, NJ: Lawrence Erlbaum Associates.

Buro, M. (2000). ORTS: A Hack-Free RTS Game Environment, *Proceedings of the International Computers and Games Conference* 2002, Edmonton, Canada.

Buro, M. (2004). Call for AI Research in RTS Games. *In Proceedings of the AAAI Workshop on AI in Games*.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55, 41-84

Byrne M. D., & Kirlik A. (2002). Integrated Modeling of Cognition and the Information Environment: Closed-Loop, ACT-R Modeling of Aviation Taxi Errors and Performance. *Aviation Human Factors Division* Institute of Aviation.

Byrne, M. D., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology, 15,* 135-155.

Cañas, J. J., Quesada, J. F., Antolí, A. & Fajardo, I. (2003). Cognitive Flexibility and adaptability to environmental changes in dynamic complex problem-solving tasks. *Ergonomics* 2003, VOL. 46, NO. 5, 482 ± 501.

Cañas, J.J., Antolí, A., Fajardo, I., & Salmerón, L. (2005) Cognitive inflexibility and the development and use of strategies for solving complex dynamic problems. Effects of different types of training. *Theoretical Issues in Ergonomics Science*. Vol: 6 (1) pp. (95-108).

Cellier, J. M., Eyrolle, H., & marine A. C. (1997). Expertise in dynamic environments, Ergonomics, 40, 28 ± 50.

Charman S., & Howes, A. (2002) The Effect of Goal Constraints on Strategy Generation. In Proceedings of the twenty-fourth annual Conference of the Cognitive Science Society. 2002.

Chung, P., & Byrne, M. (2004) Visual Cues to Reduce Errors in a Routine Procedural Task. Proceedings of the twenty-sixth annual Conference of the Cognitive Science Society. 2004.

Cooper, C. (1998) Individual differences. London : Arnold, 1998 .

Cooper, R., Fox, J., Farringdon, J., & Shallice, T. (1996): A systematic methodology for cognitive modelling. *Artificial Intelligence*, *85*, 3-44.

Cooper, R. & Fox, J. (1998). COGENT: A visual design environment for cognitive modeling. Behavior Research Methods, Instruments & Computers, 30, 553-564.

Del Missier F., Fum D. (2002) Declarative and Procedural Strategies in Problem Solving: Evidence from the Toads and Frogs Puzzle. In Proceedings of the twenty-fourth annual Conference of the Cognitive Science Society. 2002.

Demetriou, A., Christou, C., Spanoudis, G., & Platsidou, M. (2002). The development of mental processing: Efficiency, working memory, and thinking. Monographs of the Society of Research in Child Development, 67, Serial Number 268.

Dörner, D. (1996). The Logic of Failure. Cambridge: Perseus, 1996

Dörner, D., Stäudel, T., & Strohschneider, S. (1986). MORO. *Program Documentation*. Memorandum 23. Lehrstuhl Psychologie II, Bamberg Universität.

Dörner, D., & Wearing, A. (1995). Complex Problem Solving: Toward a (Computersimulated) Theory. In P. A. Frensch & J. Funke (Ed.), *Complex Problem Solving: The European Perspective* (pp. 65-99). Hillsdale, NJ: Lawrence Erlbaum Associates.

Eyferth, K., Hoffmann-Plato, I., Muchowski, L., Otremba, H., Rossback, H., Spiess, M., & Widowski, D. (1982). *Project: Organization of actions*. Berlin: Institut für Psychologie de TU Berlin (Forschungsbericht Nr. 82-4, korringierter Nachdruck).

Fincham, J.M., Carter, C. S., van Veen V., Stenger V. A., & Anderson J.R. (2002). Neural mechanisms of planning: a computational analysis using event-related fMRI. *Proc. Natl. Acad. Sci.* USA 99 (2002), pp. 3346–3351.

Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, 47, 381-391.

Forgy C. L. (1982). “Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem,” Artificial Intelligence 19 (1982): 17–37.

Franklin, A. L., & Hunt, E. B. (1993). An emergency situation simulator for examining time-pressured decision making. *Behavior Research Methods, Instruments & Computers*, 25, 143-147.

Frensch, P. A., & Funke J. (1995). Definitions, Traditions, and a General Framework for Understanding Complex Problem Solving. In: Frensch P. A., & Funke J. (Eds.), *Complex problem solving: the European perspective*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Friman, H., & Brehmer, B. (1999). Using Microworlds To Study Intuitive Battle Dynamics: A Concept For The Future. *CCRTS.*

Fu, W.-T., & Anderson, J. R. (2008). Dual learning processes in interactive skill acquisition. *Journal of Experimental Psychology, Applied*. 14 (2), 179-191.

Fu, W-T., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General*, 135(2), 184-206.

Fu, W-T., & Gray W.D. (2006). Suboptimal tradeoffs in information seeking. *Cognitive Psychology* 52 (2006) 195–242

Fu, W-T., & Gray, W. (2001). Why people often do not adopt efficient strategies? Memory versus Perceptual-Motor Tradeoffs in a Blocks World Task. *Proceedings of the twenty-third annual Conference of the Cognitive Science Society*.

Fum, D., & Del Missier, F. (2001). Adaptive Selection of Problem Solving Strategies. *In Proceedings of the twenty-third annual Conference of the Cognitive Science Society.*

Funke, J. (1991). Solving Complex Problems: Explorations and Control of Complex Systems. In Sternberg R. (ed) (1991) *Complex problem solving : principles and mechanisms*. Hillsdale. Erlbaum.

Funke, J. (1992). Dealing with Dynamic Systems: Research stretegy, diagnostic approach and Experimental results. *The German Journal of Psychology*, Vol 16 No. 1 24-43.

Funke, J. (1995). Experimental research on complex problem solving. In: Frensch P. A., & Funke J. (Eds.), *Complex problem solving: the European perspective*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Funke, J. (2001). Dynamic systems as tools for analysing human judgement. *Thinking and Reasoning*, 2001, 7 (1), 69–89.

Funke, J., & Frensch, P. (1995). *Complex problem solving research in North America and Europe: An integrative review*. University of Heidelberg. 1995.

Gerjets, P., Scheiter, K., & Tack, W. (2000). Resource-adaptive Selection of Strategies in Learning from Worked-Out Examples. *In Proceedings of the twenty-second annual Conference of the Cognitive Science Society.*

Gonzalez, C., Lerch, F. J., & Lebiere, C. (2003). Instance-based learning in real-time dynamic decision making. *Cognitive Science*, 27, 591–635.

Gonzalez, C., Thomas, R., & Vanyukov, P. (2004). The relationships between cognitive ability and dynamic decision making. *Elsevier Intelligence* 33 (2005) 169–186.

Gonzalez, C., Vanyukov, P., & Martin, M. (2005). The use of microworlds to study dynamic decision making. *Computers in Human Behavior* 21 (2005) 273–286.

Granlund, R., Johansson, B., & Persson, M. (2001). C3Fire a Micro-world for Collaboration Training in the ROLF environment. *In proceedings to SIMS 2001 the 42nd Conference on Simulation and Modelling, Simulation in Theory and Practice.* Organized by Scandinavian Simulation Society, Porsgrunn, Norway, 8-9 October.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds Matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experiment Psychology*: Applied, 6(4), 322-335.

Gray, W. D., Schoelles, M. J., & Sims, C.R. (2005). Adapting to the task environment: Explorations in expected value. *Cognitive Systems Research* 6 (2005) 27–40.

Gray, W. D., Sims, C. R., Fu, W.-T., & Schoelles, M. J. (2006). The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behaviour. *Psychological Review*, 113(3), 461-482.

Güss, C., Glencross, E., Tuason, M., Summerlin, L. & Richard, F. (2004). Task complexity and difficulty in two computer-simulated problems: Cross-cultural similarities and differences. *Proceedings of the 26th Annual Conference of the Cognitive Science Society* (pp. 511-516). Mahwah, NJ: Lawrence Erlbaum Associates.

Heineken, E., Arnold H.-J., Kopp, A., & Soltysiak, R. (1992). Strategies of thinking in controlling a simple dynamic system under different deadtime-conditions. *Sprache & Kogntion*, 11, 136-148.

Hesse, F. W. (1982). Effects of semantic context on the processing of complex problems. *Zeitschrift fur Experimentelle und Angewandte Psychologie*, 29, 62-91.

Hesse, F. W., Spies K., & Lüer, G. (1983). Influence of motivational factors on problem solving performance in interacting with complex problems. *Zeitscbrift für Experimentelle und Angewandte Psychologie*, 30, 400-424.

Hörmann, H. J., & Thomas, M. (1989). On the relation between intelligence and complex problem solving. *Sprache & Cognition*, 8, 23-31.

Howes, A., Vera, A., Lewis, R. L., & McCurdy, M. (2004). Cognitive constraint modeling: A formal approach to supproting reasoning about behavior. In proceedings of the *26th Annual Conference of the Cognitive Science Society* (pp.595-600) . August 4-7, Chicago, USA.

Howie, D. E., & Vicente K. J. (1998). Measures of operator performance in complex, dynamic, microworlds: advancing the state of the art. *Ergonomics,* 81, 485-500.

Hunt. E. (1995). The role of intelligence in modern society. *American Scientist* 83: 4.

Hussy, W. (1989). Intelligence and complex problem solving. *Diganostica,* 35, 1-16.

Hussy, W., & Granzow, S. (1987) Complex Problem Solving, memory, and kind of processing. *Zeitschrift fur Experimentelle und Angewandte Psychologie*, 34, 212-227.

Janssen, C.P. & Gray, W.D. (2012) When, What, and How Much to Reward in Reinforcement Learning based Models of Cognition. *Cognitive Science*, 36(2), 333-358.

Janssen, C.P., Gray, W.D., & Schoelles, M.J. (2008). How a Modeler's Conception of Rewards Influences a Model's Behavior: Investigating ACT-R 6's Utility Learning Mechanism. In *Proceedings of the 15th Annual ACT-R Workshop* (p. 42), Pittsburgh, PA.

Johansson B., Persson M., Granlund R., & Mattson P. (2003). C3Fire in Command and Control Research. *Cognition, Technology and Work*, 2003, Vol 5; Numb 3, pp. 191-196. Publisher Springer-verlag.

John, B., & Gray W. (1995). CPM-GOMS: an analysis method for tasks with parallel activities. *Proceeding CHI '95 Conference Companion on Human Factors in Computing Systems*. Pp. 393-394

Johnson, T. R. (1997). Control in Act-R and Soar. In M. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 343-348): Hillsdale, NJ: Lawrence Erlbaum Associates.

Jonassen, D.H. (2000). Toward a design theory of problem solving. *Educational Technology: Research & Development*, 48 (4), 63-85.

Jones, G., Ritter, F. E., & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science*, 11(2), 1-8.

Joslyn, S., & Hunt, E. (1998). Evaluating individual differences in response to time–pressure situations. *Journal of Experimental Psychology. Applied*, 4, 16–43.

Kerstholt, J. H. (1992). Information search and choice accuracy as a function of task complexity and task structure. *Acta Psychologica*, vol. 80, no 1-3 (1 p.), pp. 185-197.

Kerstholt, J. H. (1994). The effect of time pressure on decision-making behaviour in a dynamic task environment. *Acta psychologica*, vol. 86, no1, pp. 89-104 (18 ref.)

Kerstholt, J. H. (1996). The effect of information costs on strategy selection in dynamic tasks. *Acta Psychologica*, Volume 94, Number 3, December 1996 , pp. 273-290(18).

Kiefer, J., & Urbas, L. (2006). How to model different strategies in dynamic task environments. In *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 172-176). Trieste, Italy.

Kieras, D., & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*., 12, 391-438.

Kotovsky, K.; Hayes, J.R.; & Simon, H.A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology* 17, 248-294.

Kuylenstierna, J., Rydmark, J., & Sandström H. (2004). Some research results obtained with DKE: A dynamic war-game for experiments. *9th ICCRTS.* Copenhagen 2004.

LaPointe, L. B., & Engle, R. W. (1990). Simple and complex word spans as measures of working memory capacity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16, 1118–1133.

Laird, J. E. (2002). Research in Human-level AI using Computer Games, *Communications of the ACM*, 45(1), 32-35, (invited).

Laird, J. E. (2008). Extending the Soar Cognitive Architecture. *In Proceedings of the First Conference on Artificial General Intelligence* (AGI-08).

Laird, J., Newell, A., & Rosenbloom P. (1987). SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, 33.

Larkin, J. H. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. 319–341). Hillsdale, NJ: Lawrence Erlbaum Associates.

Larkin, J., McDermott J., Simon D.P., & Simon H. A. (1980). Expert and Novice Performance in Solving Physics Problems. *Science.* 20. Vol. 208. no. 4450, pp. 1335 - 1342

Lebiere, C., Wallach, D. & Taatgen, N. (1998). Implicit and explicit learning in ACT-R. In F. Ritter & R. Young (Eds.). *Cognitive Modeling II* (pp. 183-193). Nottingham: Nottingham University Press.

Lebiere, C, Anderson, J. R., & Bothell, D. (2001). Multi-tasking and cognitive workload in an ACT-R model of a simplified air traffic control task. *In Proceedings of the Tenth Conference on Computer Generated Forces and Behavior Representation.* Norfolk, VA.

Lee, F. J., Anderson, J. R., & Matessa, M. P. (1995). Components of Dynamic Skill Acquisition. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 506-511). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lee, F., & Taatgen N. (2002). Multitasking as Skill Acquisition. In *Proceedings of the twenty-fourth annual Conference of the Cognitive Science Society*.

Lee, L., & Johnson-Laird, P. (2004). Creative strategies in problem solving. In *Proceedings of the twenty-sixth annual Conference of the Cognitive Science Society*.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*. 124(1):83-97.

Lewis, R. L., Vera, A. H., & Howes, A. (2004). A constraint-based approach to understanding the composition of skill. In *Proceedings of the International Conference on Cognitive Modeling*.

Lovett, M. C. (2002). Modeling selective attention: not just another model of Stroop. Cognitive Systems Research, 3, 67-76. [special issue printing top papers from *International Conference on Cognitive Modeling*]

Lovett, M. C., & Anderson, J. R. (1996). History of success and current context in problem solving: Combined influences on operator selection. *Cognitive Psychology*, 31(2), 168-217

Lovett, M.C., & Schunn C. D. (1999). Task Representations, Strategy Variability, and Base-Rate Neglect. *Journal of Experimental Psychology: General.*Vol. 128, No. 2,107-130.

Maule, A. J., & Mackie P. (1990). A Componential Investigation of the Effects of Deadlines on Individual Decision Making, in K. Borcherding, O. Larichev, and D. Messick (eds.), *Contemporary Issues in Decision Making*. North Holland: Elsevier Science Publishers.

Mayer, R. (1998). Cognitive, metacognitve, and motivational aspects of problem solving. *Instructional Science*, 26, 49-63.

Mayer R., & Wittrock M.C. (1996). Problem-solving transfer. In: D.C. Berliner and R.C. Calfee, Editors, *Handbook of educational psychology*, Simon & Schuster Macmillan, New York (1996), pp. 47–62

Metcalfe J., & Greene M. J. (2007). Metacognition of Agency. *Journal of Experimental Psychology General*. 136(2):184-99.

Nellen, S., & Lovett, M. C. (2004). Towards a theory of balancing exploration and exploitation in probabilistic environments. In *Proceedings of the sixth International Conference on Cognitive Modeling* (pp. 214-219).

Neth H., & Payne S. (2001). Addition as Interactive Problem Solving. *Proceedings of the twenty-third annual Conference of the Cognitive Science Society.*

Neth H., Sims C., Veksler V., & Gray W. (2004). You Can't Play Straight TRACS and Win: Memory Updates in a Dynamic Task Environment. In *Proceedings of the twenty-sixth annual Conference of the Cognitive Science Society*.

Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 18:87-127.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Mass. : Harvard U.P.

Newell, A., & Simon, H. (1972). *Human Problem solving*. Prentice-Hall.

Niessen C., Eyferth K., & Bierwagen T. (1999). Modelling cognitive processes of experienced air traffic controllers. *Ergonomics,* 1999, VOL. 42, NO. 11, 1507 ± 1520.

Ohlsson, S. (2007). The Effects of Order: A Constraint-Based Explanation In Ritter, F. E., Nerb, J., Lehtinen, E., O'Shea, T. M. (Eds.) (2007). *In order to learn: How the sequence of topics influence learning*. New York: Oxford University Press.

Omodei M. M., & Wearing A. (1992). *Fire Chief Ver 2.2 User Manual.* Department of PsychologyUniversity of Melbourne.

Omodei, M. M., & Wearing, A. J. (1995). The Fire Chief microworld generating program: An illustration of computer simulated microworlds as an experimental paradigm for studying complex decision-making behavior. *Behavior Research Methods, Instruments, and Computers*, 27, 303–316.

Payne, J. W., Bettman, J. R., Johnson E. L (1988). Adaptive Strategy Selection in Decision Making *Journal of Experimental Psychology: Learning, Memory and Cognition* 14, 534–552.

Payne J. W., Bettman J.R., Johnson E. J. (1993) The Adaptive Decision Maker Cambridge University Press 1993.

Peebles, D. & Bothell, D. (2004). Modelling performance in the sustained attention to response task. In *Proceedings of the sixth International Conference on Cognitive Modeling* (pp. 231-236). Pittsburgh, PA: Carnegie Mellon University/University of Pittsburgh.

Putz-Osterloh, W., & Lüer, G. (1981). On whether results from a test of intellingence can predict problem solving performance. Zeitschrift für Experimentelle und Angewandte Psychologie, 28, 309-334.

Pylyshyn, Z. (1989). The role of location indexes in spatial perception: A sketch of the FINST spatial indexing model. Cognition, 32, 65-67.

Quesada J., Kintsch W., Gomez E. (2002) Computational Theory of Complex Problem Solving Using Latent Semantic Analysis. In Proceedings of the twenty-fourth annual Conference of the Cognitive Science Society. 2002.

Rasmussen, J. (1983). Skills, rules, and knowledge: Signal, signs, and symbols, and other distinctions in human performance. *IEEE Transactions on Systems, Man, & Cybernetics*, 15, 234-243.

Raven, J. C. (1976). *Advanced progressive matrices, Sets I and II*. Oxford, England: Oxford Psychologists Press.

Reder, L. M. (1987) Strategy selection in question answering. *Cognitive Psychology,* 19, 90-138.

Reder, L., & Schunn, C. D. (1996). Metacognition does not imply awareness: Strategy choice is goverened by implicit learning and memory. In L. M. Reder (Ed.), I*mplicit memory and metacognition* (pp. 45-78). Mahwah, NJ: Erlbaum.

Rehling J., Lovett M., Lebiere C., Reder L., & Demiral B. (2004). Modeling Complex Tasks: An Individual Difference Approach. *Proceedings of the twenty-sixth annual Conference of the Cognitive Science Society.*

Reichert, U., & Dörner, D. (1988). Heurisms in handling a simple dynamic system. *Sprache & Kognition*, 7(1), 12-24.

Rigas, G., & Brehmer, B. (1999). Mental processes in intelligence tests and dynamic decision making tasks. In: Juslin P., & Montgomery H. (Eds.), *Judgement and decision making: Neo-Brunswikean and process-tracing approaches-65*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rigas G., Carling E., & Brehmer B. (2002). Reliability and validity of performance measures in microworlds. *Elsevier Intelligence* 30 (2002) 463–480.

Ritter, F. E., Nerb, J., Lehtinen, E., & O'Shea, T. M. (Eds.) (2007). *In order to learn: How the sequence of topics influence learning*. New York: Oxford University Press.

Scheiter K., Gerjets P., & Heise E. (2000). Hypertext Navigation and Conflicting Goal Intentions: Using Log Files to Study Distraction and Volitional Protection in Learning and Problem Solving. *In Proceedings of the twenty-second annual Conference of the*

*Cognitive Science Society.*

Schoelles, M. J., & Gray, W. D. (2000). Argus Prime: Modeling emergent microstrategies in a complex simulated task environment. In N. Taatgen, & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 260-270). Veenendal, NL: Universal Press.

Schoelles M., & Gray W. (2001). Decomposing Interactive Behavior. *Proceedings of the twenty-third annual Conference of the Cognitive Science Society*.

Schoppek, W. (1991). Game and reality, reliability and validity of behavior patterns in complex situations. *Sprache & Kognition,* 10, 15-27.

Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P.J. (1997). To calculate or not calculate: A source activation confusion (SAC) model role in strategy selection. *Journal of Experimental Psychology: Learning,of problem familiarity's Memory, & Cognition*, 23(1), 3-29.

Schunn, C. D., & Reder, L. M. (1998). Strategy Adaptivity and Individual Differences. In D. L. Medin (Ed.), *The psychology of learning and motivation*. New York: Academic Press.

Schunn, C. D., & Reder, L. M. (2001). Another source of individual differences: Strategy adaptivity to changing rates of success. *Journal of Experimental Psychology: General*, 130(1), 59-76.

Schunn, C.D., Christian ,D., Lovett, M. C., & Reder, L. M. (2001). Awareness and working memory in strategy adaptivity. *Memory & Cognition,* Volume 29, Number 2, 1 March 2001 , pp. 254-266(13)

Schunn, C. D., McGregor, M., & Saner, L. D. (2006). Expertise in ill-defined problem solving domains as effective strategy use. *Memory & Cognition,* 33(8), 1377-1387.

Shah, P., & Miyake, A. (1996). The separability of working memory resources for spatial thinking and language processing: An individual differences approach. *Journal of Experimental Psychology, 1*25, 4–27.

Shortliffe, E.H. & Buchanan, B.G. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences* 23 (3–4): 351–379

Shugan, S.M. (1980). The cost of thinking. *Journal of Consumer Research* 75: 99–111.

Siegler, R. S., & Shipley C. (1995). Variation, Selection, and Cognitive Change. In T. Simon and G. Halford (Eds.). *Developing cognitive competence: new approaches to process modeling.* Hillsdale, NJ: Erlbaum, pp. 31-76.

Simon, H.A. (1990). Invariants of human behavior. *Annual Review of Psychology, 41*, 1–19.

Skinner, B.F. (1953). *Science and Human Behavior*. New York: Macmillan.

Sloutsky V., & Yarlas A. (2000). Problem Representation in Experts and Novices: Part 2. Underlying Processing. *In Proceedings of the twenty-second annual Conference of the Cognitive Science Society.*

So R., & Sonenberg L. (2004). Situation Awareness in Intelligent Agents: Foundations for a Theory of Proactive Agent Behavior IEEE/WIC/ACM *International Conference on Intelligent Agent Technology* (IAT'04) pp. 86-92

St.-Amant, R., Freed, A. R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research, 6*, 71-88.

Stäudel, T. (1987). *Problem solving, emotions and competence. The evaluation of and integrative construct.* Regensburg. FRG: Roderer

Sternberg, R. J. (1991). Triarchic abilities test. In D. Dickinson (Ed.), *Creating the future: Perspectives on educational change* (pp. 76–81). Aston Clinton, Bucks, UK: Accelerated Learning Systems.

Sternberg, R. J. (1994). *Personality and intelligence*. C.U.P.

Sternberg, R. J., & Frensch, P. A. (Eds.). (1991). *Complex problem solving: Principles and mechanisms*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stewin, L., & Anderson, C. (1974). Cognitive complexity as a determinant of information processing. *Alberta Journal of Educational Research*, 20(3), 233-243.

Streufert, S. (1997). Complexity: An integration of theories. *Journal of Applied Social Psychology*, 27, 2068-2095.

Streufert, S., Nogami, G. Y., Swezey, R. W., Pogash, R. M., & Piasecki, M. T. (1988). Computer assisted training of complex managerial performance. *Computers in Human Behavior,* 4, 77-88.

SüB, H. M., Oberauer, K., Wittmann, W.W., Wilhelm, O., & Schulze, R. (2002). Working-memory capacity explains reasoning ability – and a little bit more. *Intelligence* 20:261-288.

Taatgen, N. A. (2001). A model of individual differences in learning Air Traffic Control. In *Proceedings of the Fourth International Conference on Cognitive Modeling*, pp. 211-216. Mahwah, NJ: Lawrence Erlbaum Associates.

Taatgen, N. A. (2002). A model of individual differences in skill acquisition in the Kanfer Ackerman Air Traffic Control Task. *Cognitive Systems Research*, *3*(1), 103-112.

Taatgen, N. A. (2005). Modeling parallelization and speed improvement in skill acquisition: from dual tasks to complex dynamic skills. *Cognitive Science*, 29, 421-455.

Taatgen, N. A. & Lee, F.J. (2003). Production Compilation: A simple mechanism to model Complex Skill Acquisition. *Human Factors*, 45(1), 61-76.

Taatgen, N. A., Lebiere, C. & Anderson, J.R. (2006). Modeling Paradigms in ACT-R. In R. Sun (ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press; 29-52.

Taatgen, N. A. & Anderson, J.R. (2008). ACT-R. In R. Sun (ed.), *Constraints in Cognitive Architectures*. Cambridge University Press, pp 170-185.

Taatgen, N. A., & Anderson, J.R. (2010). The Past, Present, and Future of Cognitive Architectures. *Topics in Cognitive Science 2* (2010) 693–704

Thunholm, P. (2005). Planning Under Time-pressure: An Attempt Toward a Prescriptive Model of Military Tactical Decision Making. In, H. Montgomery, R. Lipshitz & B. Brehmer (Eds.), *How Experts Make Decisions*. New Jersey: Lawrence Erlbaum.

Valax, M. F., & Cellier, J. M. (1992). Aid for the organization of work in the workshops: the gap between problems and production forecast. In G. de Terssac, P. Dubois (Eds.), *Les nouvelles rationalisations de la production* (pp. 121-137). Toulouse: Cepadues.

VanLehn, K. (1989). Discovering problem solving strategies: What humans do and machines don't (yet). In A. Segre (Ed.), *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 215-217). Los Altos, CA: Morgan Kaufmann.

Veksler, B. Z., Gray, W. D., & Schoelles, M. J. (2007). From 1000ms to 650 ms: Why interleaving, soft constraints, and milliseconds matter. In *proceedings of the 8th International Conference on Cognitive Modeling*. Ann Arbor, Michigan, USA.

Von Neumann, J., & Morgenstern, O. (1947). *Theory of games and economic behavior*, 2nd ed. Princeton, NJ: Princeton University Press.

Wenke, D., Frensch, P. A., & Funke, J. (2004). Complex problem solving and intelligence: Empirical relation and causal direction. In R.J. Sternberg & J. Pretz (Eds.), *Cognition and Intelligence*. New York: Cambridge University Press.

# Appendix A: Publications

De Obeso Orendain, A. & Wood S. (2012). An Account Of Cognitive flexibility and inflexibility for a complex dynamic task. *Proceedings of the 12th International Conference on Cognitive Modeling*. Berlin, GER : Drexel University. Technische Universität Berlin.

De Obeso Orendain, A., & Wood, S. (2010). Cognitive Modeling of Strategies in Dynamic Tasks. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 52-57). Philadelphia, PA: Drexel University.