Luís Morais Sarmento

An Emotion-Based Agent Architecture

Tese submetida à Faculdade de Ciências da Universidade do Porto
Para obtenção do grau de Mestre em Inteligência Artificial e Computação
sob orientação do Professor Eugénio de Oliveira

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Março / 2004

aos meus pais Maria Helena e Luís Carlos

AGRADECIMENTOS

Este trabalho nunca teria sido possível sem a ajuda e contribuição a vários níveis de pessoas que certamente não conseguirei nomear na totalidade.

Em primeiro lugar, gostaria de agradecer ao Professor Eugénio de Oliveira pelo confiança depositada em mim, pelo constante desafio que me colocou, pela forma esclarecida e objectiva como ia levantando os problemas e apontando os caminhos a seguir, e por me ter permitido expandir os meus horizontes em variadas direcções.

Gostaria de agradecer a toda a equipa do Núcleo de Inteligência Artificial Distribuída e Robótica (NIAD&R) da Faculdade de Engenharia pelo constante encorajamento, pelos comentários e ideias e pelo agradável ambiente de trabalho que sempre proporcionaram. Gostaria de agradecer em particular ao Daniel Moura, com o qual tive o prazer de colaborar, pelas inúmeras discussões e trocas de ideias sobre Agentes Emocionais.

Relativamente à execução do relatório gostaria também de agradecer à Cristina Sá, pela ajuda na paginação e execução gráfica e pelo longo encorajamento que me proporcionou, e também à Ana Sofia Pinto, pela ajuda na edição em inglês e pela inesgotável fonte de boa disposição.

Gostaria de deixar também uma palavra de agradecimento à Professora Belinda Maia e à Professora Diana Santos pela paciência e pelo encorajamento e suporte que me deram durante este tempo que tenho pertencido à equipa da Linguateca.

Agradeço a todos aqueles que durante execução deste trabalho tiveram a paciência de me apoiarem incondicionalmente e aos quais eu fui roubando o tempo e a energia que me mereciam: Cristina, Ricardo, Sahra e tantos outros. Agradeço em especial à Teresa toda a força que me dá.

Por fim, agradeço aos meus pais, ao meu irmão, à minha avó e à minha restante família por todas as coisas que estas páginas não chegariam para descrever.

The question is not whether intelligent machines can have any emotions,
but whether machines can be intelligent without any emotions.

The Society of Mind - Marvin Minsky

RESUMO

A Emoção Humana estabelece um conjunto de interacções funcionais com a Cognição. Ao contrário do que foi durante muito tempo defendido, estas interacções foram identificadas como essenciais para a emergência de comportamento inteligente em ambientes complexos e com requisitos de tempo-real. Este trabalho procura introduzir em Arquitecturas de Agentes Autónomos mecanismos com *funcionalidades* semelhantes aquelas que as Emoções possuem na Cognição Humana, de forma a permitir melhorar o comportamento de Agentes Autónomos em ambientes complexos, próximos da realidade que nos rodeia. Tal arquitectura poderá ser denominada de Arquitectura de Agentes baseada em Emoções.

Começaremos por discutir a motivação para este trabalho assim como colocar o nosso trabalho numa plano mais vasto que cruza a Inteligência Artificial, a Psicologia e a Neurologia. Em seguida será feito um resumo das propriedades funcionais da Emoção Humana que tomamos como base para a construção de um modelo de Mecanismos Emocionais. Iremos também apresentar algum do trabalho que tem sido desenvolvido por investigadores da Inteligência Artificial com objectivos semelhantes ao nosso, e que nós consideramos mais relevante e inspirador.

Em seguida iremos apresentar um modelo genérico de Mecanismos Emocionais que permite a implementação de 4 propriedades funcionais da Emoção, nomeadamente: (i) a utilização da Emoção como Informação, (ii) a utilização da Emoção como mecanismo de Controlo de Processos Cognitivos, (iii) a Emoção como mecanismo para a Distribuição de Recursos Computacionais e (iv) como método para definição de uma Estratégia Global de Processamento de Informação.

Seguidamente, apresentaremos o simulador de fogos florestais Pyrosim, que serviu de base para o desenvolvimento da Arquitectura de Agentes baseada em Emoções. Esta plataforma foi desenvolvida de raíz e permite simular um ambiente em que os Agentes cooperam para combater um fogo florestal. A complexidade e os requisitos de tempo-real do Pyrosim tornam evidente a necessidade da utilização de Mecanismos Emocionais, proporcionando desta forma uma base conveniente para o desenvolvimento de uma Arquitectura de Agentes baseada em Emoções.

Prosseguiremos com a apresentação da Arquitectura, que se encontra adaptada ao ambiente proporcionado pelo simulador Pyrosim. Serão descritos os módulos que a compõem e a forma como estes são divididos em duas camadas sobrepostas. Será explicada a forma como os Mecanismos Emocionais estão integrados nesta Arquitectura e o modo como permitem suportar as propriedades funcionais das Emoções Humanas. Explicaremos como 3 Mecanismos Emocionais concretos, “Medo”, “Ansiedade” e “Auto-Confiança”, foram instanciados na Arquitectura e proporcionam uma vantagem efectivam em termos de adaptabilidade e gestão de recursos computacionais. Serão explicadas interacções funcionais (a alto e a baixo nível) entre os Mecanismos Emocionais e os módulos funcionais específicos da Arquitectura e a forma como elas se repercutem nos comportamentos de alto nível.

Finalmente serão apresentadas as conclusões deste trabalho e algumas possibilidades de trabalho futuro.

ABSTRACT

Human Emotion establishes a set of functional interactions with Cognition. Contrary to what was advocated for a long time, these interactions were identified as essential for the emergence of intelligent behavior in complex environments with real-time requirements. The present work strives to provide Autonomous Agent Architectures with mechanisms with *functionalities* similar to those Emotions have in Human Cognition. This would enhance Autonomous Agents behavior in complex environments, as the ones we live in. Such Architecture may be called an Emotion-based Agent Architecture.

We will begin by discussing the motivation behind this work as well as trying to place it in a wider context, which crosses Artificial Intelligence, Psychology and Neuroscience. Then, we will make a summary of Human Emotion's functional properties, which were the basis for building a model of Emotional Mechanisms. We will also present some of the work that has been developed by AI researchers with goals similar to ours, which we consider most relevant and inspiring.

The introduction of a general model of Emotional Mechanisms will follow. This model enables the implementation of four functional properties of Emotion, namely: (i) Emotion as Information, (ii) Emotion as a Process Control Mechanism, (iii) Emotion as a Resource Allocation Mechanism, and (iv) Emotion as a Global Information Processing Strategy.

Next, we shall introduce the forest fire simulator Pyrosim, which was the basis for the development of the Emotion-based Agent Architecture. This platform allows us to simulate an environment in which Agents cooperate in order to fight a forest fire. The need to use Emotional Mechanisms is made very clear by Pyrosim's complexity and real-time requirements. Thus, Pyrosim becomes a convenient basis for the development of an Emotion-based Agent Architecture.

We will proceed by introducing the Architecture, which is adapted to the environment created by the Pyrosim simulator. A description of both the Architecture's modules and the way in which they are divided in two layers will follow. We will explain how Emotional Mechanisms are integrated in this Architecture and how they are able to support the functional properties of Human Emotions. Then, we will explain how three actual Emotional Mechanisms, "Fear", "Anxiety" and "Self-Confidence", were instantiated in the Architecture. We will also show how the instantiation of such Emotions results in a real advantage in terms of adaptability and of computational resources management. We will explain the functional interactions (high and low-level) between Emotional Mechanisms and the specific functional modules of the Architecture. An explanation of the way in which these interactions reflect on high-level behaviors will also be given.

Finally, we will present the conclusions of this work as well as some ideas for future work.

TABLE OF CONTENTS

CHAPTER 1. GENERAL INTRODUCTION

1.1. Introduction

In this thesis report we introduce an Agent Architecture that includes a set of Emotional-like Mechanisms[1] that deeply interact with the operating modes of the Agent similarly to the way Emotion interacts with Human cognition. These Emotional-like Mechanisms try to functionally adapt the Agent's information processing capabilities in order to improve its global performance in the environment. We follow the view of those arguing that these Emotional-like mechanisms become a requirement for intelligent behavior for resource-bounded Agents operating in complex, dynamical and real-time environments.

According to several theories developed in the fields of neuroscience and psychology, Emotions are crucial for the feasibility of the Human intelligence apparatus in the real world [Dam94][Fri86]. Humans are undoubtedly resource bounded-agents with multiple simultaneous goals. Nevertheless, they are able to live with some degree of success in very complex environments, performing actions in real-time. Emotions indeed participate in the intelligence processes required for this success. Following this line of reasoning, we have developed an Agent Architecture that includes mechanisms functionally analogous to Human Emotion but applied to Synthetic Agents.

As we will try to show in the following Chapters, Emotion is a complex subject that crosses several fields of knowledge including Psychology, Neuroscience, Artificial Intelligence and Philosophy. In each of these fields, one will most certainly find many different perspectives on the study of Emotion and different schools of thought, whose ideas are sometimes opposing or difficult to conciliate. Therefore, whoever starts a study in this field faces not only a large amount of almost contradictory literature, but also an ideological and terminological nightmare. The ideas supporting this thesis are rather dispersed over such literature. In this thesis, much of the effort has been devoted to mining the field in order to build a coherent view of the subject. However, we must emphasize that along this process there has always been an effort to keep an engineering view about Emotion. This means that each idea or model used or developed would always have to make sense from a functional

[1] Throughout this thesis the term "Emotion" will be applied only when referring to Human Emotion. We will use the term "Emotion-like Mechanism" or simply "Emotional Mechanism" when mentioning any computerized model or implementation based on Human Emotion.

point of view. This way, we obviously come closer to the perspective of the *Functionalists,* with all the inherent advantages and disadvantages [Mof00]. A detailed discussion about the Functionalities of Emotions will be developed in Chapter 2.

In the following sections of this chapter we will explain the motivation behind this work. We will also try to place our work in the wider panorama of the study of Emotion within Artificial Intelligence or, more generally, Computer Science. We will also make a very brief description of our testing scenario, the Pyrosim environment, through which we have tested our Emotion-based Architecture. Finally, we will present the outline of the main body of this thesis.

1.2. Motivation

The beginning of this work has been greatly inspired by Damásio's ideas about the role of Emotion and Feelings in the rational processes of intelligence, described in the book "Descartes' Error" [Dam94]. Although Damásio has not been the first author to address these interactions, his book has had the merit of attracting both the specialized audience and the general public to the subject of Emotion and, therefore, the merit of starting a wide debate about this subject.

Our interest in Emotion came in a rather natural way after reading Damásio's book. The ideas described in it suggested the existence of some very efficient engineering processes performed by the Emotional mechanisms. The connection between Emotion and decision-making and Emotion and planning, described by Damásio, immediately drew our attention, as both these capabilities are central to the field of Autonomous Agents. The implication that Emotion would not only help Humans on deciding and planning their actions *rationally* but also that its absence would render most of the decision-making and planning processes unfeasible in the real world, seemed very intriguing. At the same time, these ideas became very appealing if we considered that Humans, despite of the limited intellectual and physical resources they have available, are reasonably successful Autonomous Agents: they are capable of operating in real-time in a very complex, dynamic and uncertain environment, the real world. Since the "real world" is the ultimate application scenario for Autonomous Agents, it was certainly worth looking at Emotion, whatever this could be, as a possible interesting and useful contribution to Agent Architectures.

The relationship between Emotion and decision-making became even more interesting if we considered that the Human brain is a highly distributed and massively parallel system. Each intelligence-based process involves several brain (and also body) components that work both in parallel and cooperatively in order to produce a final result. For example, a decision-making process would require at one stage a method for assessing the current situation as well as detecting which features are most relevant for the decision. This would require both the analysis of the state of the world and of one's previous experiences. Thus, a mechanism optimized to access memory for "relevant" information according to the current situation would also be needed. At the same time, one would draw a projection in the future of several possible courses of action, based on past experiences and also according to the notion of how the world may change after our actions. The evaluated options would take into account one's personal interests and preferences, social rules and norms, the notion of one's own capabilities and several other criteria. If we now consider that each of these sub-processes is itself complex, depending on multiple parameters, and possibly making use of other complex capabilities running in parallel, we may conclude that trying to obtain a coherent response from all elements involved is a major coordination challenge. Achieving it, is a huge *engineering* accomplishment.

Damásio pointed out that Emotion plays a very significant role in the success of these type of interactions, for example by speeding up memory searches using Emotional cues, by reducing the space of analysis to a set of reasonable hypothesis, or by providing a way to compensate the uncertainty of data gathered by perception. As engineers working in the field of Autonomous Agents, it was obvious for us that there was a strong engineering potential in several of the concepts described by Damásio for the development of better Agent Architectures.

Most of the present proposals for Agent Architectures have been inspired, at least remotely, on the Human brain and explore the knowledge gathered about its structural and functional properties. However, it is rather surprising to observe that until very recently few were the cases where Emotion-based concepts were clearly included as key concepts of the Agent Architecture. Some notable exceptions include the work of Herbert Simon [Sim67] and Aaron Sloman [Slo81] who have pioneered this field and whose work and ideas have been great sources of inspiration for more recent researchers.

Several reasons may explain this apparent lack of interest in Emotion. The first one is related to the common and long debated misconception that Emotion is the opposite notion of Reason. Acting Emotionally has always implied that one's action would be impulsive and unthought, and therefore always less appropriate or efficient than action taken after a long Reasoning process. Emotion and Reasoning have historically been placed on opposite fields. The only interaction Emotion was considered to have with Reasoning was clearly of a destructive nature. The other reason for the absence of Emotion-based concepts in Agent Architectures is quite basic: we simply did not know enough about Emotion to build a comprehensive model. It was not until the mid 80's that researchers had the chance to observe the Brain performing in real-time using very powerful image acquisition and visualization techniques such as CAT, MR and PET [Mau93]. These techniques allowed researchers to thoroughly analyze the structure of the brain and to visualize its activity patterns and internal interactions, providing deeper insights into Emotion machinery.

In spite of these undeniable advances, Emotion is still a big mystery. Many basic questions remain to be answered, including, for example, the concise definition of Emotion itself. The nature of Emotion is still under intense debate [Ekm94] and there is no assurance that a definite answer will ever be found. There are still many questions to be answered. We will try to address some of them in the next chapter, which is completely dedicated to Emotion.

One issue, however, may now be considered to be completely settled in this debate: it is no longer possible to draw a complete picture of Intelligence excluding the processes related to Emotion. It is precisely at this point that artificial intelligence researchers reenter the scene, nearly 30 years after the work developed by Herbert Simon. When Agent Architectures are applied in the development of intelligent Robots, researchers find similar implementation problems to those that would be found in the hypothetical design of the Human brain. In fact, Autonomous Robots share many design requirements with Humans (although simplified). Robots should be able to operate in complex environments, subjected to some time constraints, mechanical limits, energy storage limits and, of course, limited computational power. At the same time, the world outside the Robot is complex, noisy, constantly changing and demands attention to address multiple simultaneous needs or *concerns*.

During the last decade, researchers have produced a considerable number of interesting works in which Emotional-based concepts were applied in Agent Architectures. We will make a revision of some of these works in Chapter 3. Still, it is worth mentioning some authors whose work has greatly influenced our ideas, either by their functional and practical perspective or, on the other hand, by their high-level and integrating approach. For example, Velásquez [Vel98] has applied Emotional-based mechanisms to control behavior selection of a physical robot named Yuppi. Cañamero [Cañ97][Cañ00] has also developed some very interesting work in modeling and applying Emotion-based concepts to action-selection in software Agents. Gadanho & Hallam[Gad98] and Gadanho[Gad02] built a simulation where they have applied Emotional Mechanisms to perform reinforcement learning. From a more theoretical point of view, Aaron Sloman and his students in the Cognition and Affect Group, University of Birmingham [Wri97][Bea94][Slo00] followed a *design-based approach* to develop Agent Architectures. These Architectures were based on the study of the requirements needed for building systems with Human-like intelligence. In such Architectures, Emotion-like manifestations would arise in the system as a natural result of interactions established between architectural components.

Besides their strong scientific contribution, these works, among others, had the merit of bringing the study of Emotion back to AI, after the initial works developed in the 60's. However, the complexity surrounding Emotion has, most of the times, led AI researchers to focus on very specific Emotion-Cognition interactions such as, for instance, interactions between Emotion and Learning or between Emotion and Action-Selection. In our opinion, this specificity, although almost unavoidable at that stage, has had three undesirable effects regarding the development of Autonomous Agent Architectures:

1. the over-fitting of the models of Emotion-Cognition interaction, which prevented their easy transposition to other application scenarios or even to other possible Emotion-Cognition interactions within the same scenario;

2. the inclusion of very human-specific elements in the model, such as hormones and other “organic” elements, whose direct application in the context of Synthetic Autonomous Agents, despite of its metaphorical nature, may lead to rather artificial concepts.
3. a great fragmentation of models of Emotion-Cognition interaction within the realm of Autonomous Agent Architectures, and the lack of a general, large scope model;

Another thing that seemed apparent in the early steps of our work is that AI researchers tend to choose one particular “school of thought” about Emotion and do not usually consider complementing the models they develop with ideas from others “schools”. This is not to be blamed because, obviously, one should not expect AI researchers to develop a complete model of Emotion when researchers from Psychology and Neurophysiology are still debating very fundamental issues. Nevertheless, in this thesis we have tried to follow some of the latest trends in psychology [Sch00][Bow00]: that try to integrate previous contributions under a more global model. Thus, we have taken the liberty of borrowing concepts from different “schools of thought” whenever we felt that they had a strong functional use.

Having said all this, we may now present more clearly the motivation for our work, which includes essentially 3 points:

1. *The study of Emotional-phenomena* from a very functional point of view, trying always to explore possible practical applications in Autonomous Agent Architectures.
2. *The development of a model of Emotional-based mechanisms* compatible with various functional models of Emotion dispersed among artificial intelligence, psychology and neurosciences.
3. *The development of an Agent Architecture* where such Emotional-based mechanisms provide a performance advantage, especially for resource-bounded agents that operate in complex and dynamic environments.

1.3. Some Previous Remarks

It is important to make a few considerations about the motivations presented before.

Firstly, during this work we will make a great effort to develop a clear understanding about Emotion-Cognition interactions but our focus will be exclusively on possible functional aspects. This implies that a large number of Emotion-related phenomena (cognitive or physiological) will simply not be addressed. However, this also motivates us to try to filter out those aspects of Emotion that could be ignored in a first approach when thinking of Synthetic Agents. For example, when dealing with the physiological aspects of Emotion, we have tried not to transpose them directly to Synthetic Agents. Instead, we have tried to adapt them or generalize them. This adaptation follows the idea of

Embodied Intelligence [Dam94] where the Human body is itself a part of the global intelligence machine.

Therefore, the "body" of a synthetic agent may have *functionally analogo*us interactions with Emotion but will certainly have different implementations. For example, and just for the sake of (naïve) illustration, if the Human heartbeat is accelerated in response to a fearful situation in order to prepare the body for a possible powerful response, in the case of a Synthetic Agent this could correspond to speeding up the processor clock to cope with a "dangerous" flow of data, or boost the global performance of the Agent. On the other hand, if that heartbeat acceleration is triggered by a hormonal discharge, which functions as a global communication mechanism between the brain and the body, then, from a functional point of view, it might not make much sense to try to introduce the idea of a "synthetic hormone discharge" because a similar communication mechanism can be simply performed by means of a hardware or software interrupt. Note, however, that in both cases it would probably not be possible to sustain such a powerful state for a long time either due to certain physical limits, as in the case of Humans, or to possible energy and temperature constraints in the case of Synthetic Agents.

Even by reducing the scope of our work to the study of functional interactions, we do not claim that we will try to develop a *global* model of functional Emotion-Cognition interactions, applicable in all cases to Synthetic Agents. That would be out of the scope of this kind of work. In spite of that, by trying to look at Emotions as *abstract functions of intelligence*, whose "implementation" in humans and animals is tightly related to specific physical and mental limits, one may find principles and rules of design that can be applied in the construction of Synthetic Agent Architectures. That has been the aim of our effort.

1.4. More on Emotion and Computer Science

Before proceeding with the discussion regarding how Emotions may interact functionally with Agent's internal processes, it might also be interesting to look at other perspectives on Emotion within the field of AI and Computer Science. In fact, the complexity of Emotion has motivated a great amount of work where Emotion and Emotion-based concepts play different roles in a computer system. Besides the perspective we are following, there are at least three different ones that should be considered:

1. *Recognition of Human Emotion*
2. *Expression of Emotional-like behavior*
3. *Modeling and Simulation of Human Behavior*

The categories listed here should be seen only as a reference for the major concern of researchers. In practice, actual projects may fit into more than one category as these subjects tend to blend in. However, they represent perspectives on Emotion significantly different from the one we have been pursuing. The purpose of the following section is just to emphasize that difference.

1.4.1. Recognition of Human Emotion

Emotion recognition is a basic feature of human intelligence. By recognizing Emotions, people extract additional information that may not be conveyed directly by speech, gesture, text or other forms of explicit communication. Emotion recognition plays a central role in decoding signals during a conversation and helps to reduce the possible ambiguity of a message.

People usually communicate large amounts of Emotional information, either intentionally or unconsciously. The message becomes not only "what" is transmitted but also "how" it is transmitted. One might consider two parallel communication channels: along with a "plain" information channel, people use what we may call an "Emotional information" channel.

Emotions influence our body movements and expressions. This effect is known as *Sentic Modulation* [Pic94]. Our motor system may be seen as the carrier of Emotional responses in the same way a high frequency sinusoid is the carrier of an AM audio signal. Sometimes, such bodily messages might be easily perceived while in other cases they might be expressed in such a subtle way that they can only be detected using special equipment.

Table 1.1 Effects of Sentic Modulation (taken from [Pic94])

Apparent effects	**Less apparent effects**
Facial expression Voice Intonation Gestures and Body Posture Pupilary Dilatation	Respiration Heart rate, pulse Temperature Electrodermal Fluctuation Blood Pressure

Nowadays, the richness of this information is simply discarded by computers. The most common way of transmitting information to a computer is either by using simple keyboard devices, which reduce messages to a stream of well-formatted characters, or through the clicks of a mouse or other similar pointing devices. Obviously, this state reduction seriously decreases the possibility of Human-Machine Interaction but it also reduces the possible intensity of communication between Humans made through computers.

Certainly, computers able to recognize the user's Emotional states would boost interaction possibilities. Such computers would be equipped with a wide variety of sensor systems ranging from video cameras, speech recognition modules with intonation analysis, body temperature sensors, blood pressure and electrodermal measurement devices, as well as other capture devices. Computers would then be able to detect the user's Emotional state and react accordingly. Computers could react to user frustration or disorientation (expressed, for example, through facial expression and body posture). Such a feature would be most useful in Interactive Tutoring Systems and, of course, in Human Computer Interaction in general [Zho00][Ell99]. Other interesting possibilities include having computers decoding our Emotional responses during a computer-mediated communication (chat, e-mail...) and recreating the same Emotional behavior in the receiver terminal (for example, using a 3D character).

There are several very complex implementation issues in this field. These issues range from building effective physical sensors, to extract usable information from the user, to developing algorithms that analyze that data and choose the best response from the system (possibly after a learning period). For example, recognizing Human Emotion from videos of the users is not currently feasible both because of the computational power required but also because it demands very advanced Human face recognition techniques (the interested reader is invited to [Zha00] and [Bar99]).

Apart from these extremely complex technical implementation issues, there are several important aspects regarding Emotion recognition that will eventually have to be considered, although possibly only in a distant future:

- *Sentic Modulation* can be controlled or simulated and even people may have a hard time recognizing other people's Emotions. Will computers be able to do it, or will they be easily fooled?
- On the opposite corner, there is a chance, although remote, that computers will become more efficient than people in recognizing Emotions. What possibilities does this bring?
- Do we want to have a computer recognizing our Emotions and use them as *programmed* by another person?

These are very motivating questions that, however, are beyond the scope of our work.

1.4.2. Expression of Emotional-like behavior

Remarkably, computer expression of Emotion is, somehow, a less problematic issue. Since we were kids we have been in contact with objects and images created to "express" Emotion as if they were actually feeling that Emotion. Cartoons and animation films are a good example of how completely synthetic characters can effectively express and communicate Emotion and behave in an Emotionally believable way.

There is a huge amount of experience in this field that can easily be adapted to computers, especially with the help of visual agents, usually called *Synthetic Characters*. A very famous example of a Synthetic Character capable of expressing Emotional-like behavior is the Microsoft Agent technology [Mic04]. A reasonable set of Emotional expressions may be simulated using small 2D animation sequences triggered by user and system events. Such a Synthetic Character can effectively express a wide range of Emotional states such as surprise, anger, happiness or virtually any other Emotion that can be expressed by a traditional 2D cartoon.

Other more advanced systems involve using 3D models to create Synthetic Characters. 3D techniques have the advantage of allowing developers to create realistic characters whose appearance can be parameterized, allowing almost complete control over face expression, body posture and movements, in real-time if needed. Interesting examples of these systems include EMOTE [Cos00] or the system developed at MIT and described in [Rus99]. This type of technology is now widely established among developers and researchers. In fact, very similar technology is currently used in the computer game industry, and several vendors provide complete software packages that can be adapted for this purpose [Bos].

The major reason for trying to develop Synthetic Characters capable of expressing Emotions is concerned with believability [Bat94]: characters that act Emotionally and whose appearance changes with Emotion become more real, more engaging and look as if they were alive. As pointed out in [Kli99], the expression of Emotions helps to reveal a character's inner side and to induce certain expectations on the user. Therefore, Synthetic Characters become particularly believable if their behavior conforms to those implicit expectations. For a good theoretical discussion concerning Believability and Emotions refer to [Ort01].

The issue of Believability becomes particularly important if we think people tend to spend a very significant amount of their time working with or using a computer. Synthetic Characters could help to increase the quality of such time not only by providing a user-friendlier interface but also by "humanizing" the interaction patterns between man and machine.

It is important to note that although Synthetic Characters' global behavior and visual appearance may look very complex, the underlying model of Emotions may not be as complex. In fact, a set of rules that maps events and conditions to plausible Emotional States[2] and another set of rules that maps back Emotional States to behaviors (or changes in the visual appearance) may be enough to provide the Synthetic Characters with an Emotional-like behavior. Naturally, these rules are integrated in the global architecture that is usually complex and includes additional mentalistic concepts (drives, goals, beliefs, etc) and other advanced capabilities that provide greater depth to the Agent. For example, in order to adapt themselves to users or to their virtual environment, many of these Synthetic Characters require some learning capabilities. It is interesting to note that such learning methods need to cope with a complex environment as well as being able to deal with the existence of certain *intentions* or *personality trai*ts of the Agent [Yoo00].

Most of the work developed in the field of Synthetic Agent and Virtual Environments concerning Emotion has been based on simplified versions of the model of Emotions proposed by Ortony, Clore and Collins [Ort88], which is very convenient for rule representation. We will refer to

[2] For now, let us assume that an Emotional State is simply a vector containing the values of certain internal variables.

the OCC model in the next Chapter. Briefly, this model argues that the Emotional state of an Agent results from its appraisal of the situation regarding the consequence of events, Agent actions, other Agents, and objects' aspects. From an engineering point of view, this process of Emotional elicitation may be implemented using a set of rules relating various aspects of the environment (events, other Agents, outcome of actions) to a possible Emotional state.

Besides Synthetic Characters, where the expression of Emotion is usually evident because of its graphical representation, there are other important applications for the artificial expression of Emotion. Considering that society is becoming increasingly automated, it may not be too farfetched to imagine that in a near future Autonomous Agents will probably be producing analysis and reports about Humans (e.g.: expert systems). In such a scenario, Agents will not only have to produce accurate results but will also have to be capable of communicating them to users in a personal way. This will probably require computers to express Emotions by conveying some affection in communication, as a Human would expect. In this context, developing systems capable of generating natural language and express Emotion becomes a major concern and it is a huge challenge already being tackled by some researchers. For example, in [Ros99] the authors present a system that is able to generate medical text reports using a discourse that expresses some affection to the patient. The system is thus capable of producing reports closer to those produced by Human doctors. For a comprehensive bibliographic revision about Natural Language Generation and Emotions, the reader is invited to [Piw02]

There are still other ways for Agents to express Emotions. For instance, voice intonation is also a powerful way of expressing Emotions. *Sentic Modulation* has a very strong influence in voice and such factor can also be used to transmit recognizable Emotional information. Speech generation systems are now starting to include these features but the surrounding complexity still holds back a full implementation. For additional information on this subject the interested reader should refer to [Cah90] [Fle02].

A good and in-depth discussion on most of the issues presented here may be found in [Cas00].

1.4.3. Modeling and Simulation of *Human* Behavior

Although there is a significant overlap between the different approaches to Emotion, it is still possible to consider a separate category for those research projects that deal specifically with modeling and simulating *Human* Emotion. It is not very easy to individually distinguish the projects belonging to this specific category from the others, but we will try to ground that separation on the fact that, in this case, researchers try to develop systems based on deep models of Human Emotion. The focus of their research is not to create *believable* characters (not necessarily Human-like) or systems capable of perceiving the user's Emotional state but, instead, to create computer representations of Human behavior that are both plausible and *realistic*. Researchers have acknowledged that in order to be possible to achieve such a realistic behavior, deeper models of Human Emotion must be embedded in Agent Architectures.

Deeper models of Human Emotion are essential whenever the goal is to build very realistic training environments where participating Agents simulate Human behavior and interact with users. Emotion participates in modulating the Agent's visual appearance and in its decision-making processes, but fundamentally in providing a greater depth and even a certain degree of unpredictability to its behavior, which is typical of Human interactions and therefore enhances the realism of the training environment. Obviously, in order to produce a globally realistic simulation environment, many of the visualization techniques described in the previous section have also to be applied, which simply confirms the enormous overlap between all these approaches.

It is impossible to make a complete revision of the work developed in this area so far because much research, mostly motivated by military or security concerns, is currently being produced. However, we will try to illustrate the previous discussion with two examples.

In [Gra01], Gratch and Marsella extended the Mission Rehearsal Exercise (MRE) system, a military simulation environment, by combining two complementary models of Human Emotion that they had previously developed, *Émile* [Gra00] (Emotional elicitation) and IPD [Mar00] (effect of Emotion on behavior) with the Agent Architecture of the MRE system. Demonstrations of the system to experienced military personnel indicated that the behavior of simulated characters (locals or friendly and hostile forces) became richer and more realistic. Nevertheless, authors felt that it was difficult to assess how much improvement had been obtained in training effectiveness.

Another work in this field was presented in [Sil02]. The authors developed an Agent Architecture based on an extended version of the OCC model used to simulate the behavior of crowds. Emotion was used to alter the Utilities Agent dynamically and influence its decision-making processes. A particularly interesting point about this work is that the influence of Emotions at the individual level has a decisive role in simulating social interaction between different groups of Agents, motivating the individual Agent to join a specific group and participate in its activities.

1.5. The Pyrosim Simulation Platform

We would now like to briefly present the simulation platform that we have developed in order to test and refine our ideas regarding Emotional Agent Architectures. Right from the beginning of this project we felt that it was important to have a concrete scenario upon which we would build the Agent Architecture. Despite the fact that the main focus of this work is the development of an Emotion-based Agent Architecture, we have dedicated a considerable amount of effort in building a complete simulation platform to support the software Architecture development.

The decision of developing a simulation platform from scratch was made after considering several of the publicly available simulators. In fact, none of the analyzed simulators seemed appropriate for our goals. In most cases, simulators involved very simplistic scenarios that did not require the Agents to have capabilities complex enough to justify an Emotion-based Architecture. In other cases, although the simulation was complex enough, we did not find the domain sufficiently appealing either because it was a *toy* scenario or, on the other hand, because it involved simulation of wars or military operations, which we find very unattractive.

We thought it was important to base the development of our Architecture working a problem that is closer to our own reality. After some discussion about possible domains, we decided to develop a simulation of a forest-fire combat environment. The simulator was baptized Pyrosim, from *Piros* (ancient Greek for *fire*) Simulator. The complete simulation platform is composed of three elements (the simulation server, the AgentSkeleton layer and the Pyroviz Visualizer) and it is named Pyrosim Simulation Platform (PSP).

The Pyrosim platform simulates a forest-fire scenario in which a team of Agents (firemen) cooperates to control and extinguish the fire, while simultaneously trying to minimize the overall damage and losses. In Pyrosim, Agents have to deal with dynamic fire-fronts, terrain constraints and their own physical and logistic limitations. Factors such as wind intensity, vegetation density and terrain slopes, are taken into account in simulating the progression of fire-fronts.

Each Agent needs to ensure its physical safety while trying to combat the fire and, at the same time, to help colleagues to remain safe. Agents are equipped with a limited power water jet that allows them to extinguish the fire, but they are usually not able to do it individually, so there is an obvious need for cooperation. Agents may communicate with each other in order to organize team efforts (broadcast and 1-to-1 messages). Agent's Perception System provides information about its own state (physical energy, speed, acceleration, position in the terrain, status of the personal water jet) as well as several matrix structures named *Vision Maps,* which describe close range and medium range surroundings. Visual Maps contain information with different *levels of detail* and *noise* (depending on the distance) about terrain, vegetation, level of destruction, and fire cells. Agents also receive information about visible parameters of other Agents (location, approximate energy level and action).

The simulation runs on a server application (developed in C++/Linux) that accepts network connections from Agents. Agents can run on different computers and interact with the server via socket connections. We have developed a specific layer - the *AgentSkeleton* - that deals with all low-level details between the server and Agents, providing the developer with a medium-level interface upon which Agents may be more easily programmed.

The Pyrosim platform also provides a tool for monitoring the simulation's state: the Pyroviz. The Pyroviz allows developers to visualize in real-time what is happening during the entire simulation through a 3D rendered scene, as well as track all the communication activity between the participating agents. We found this application very helpful in the development and testing of our Architecture.

The following figure is a snapshot of the Pyroviz interface showing the Pyrosim scenario together with some Agents in action.

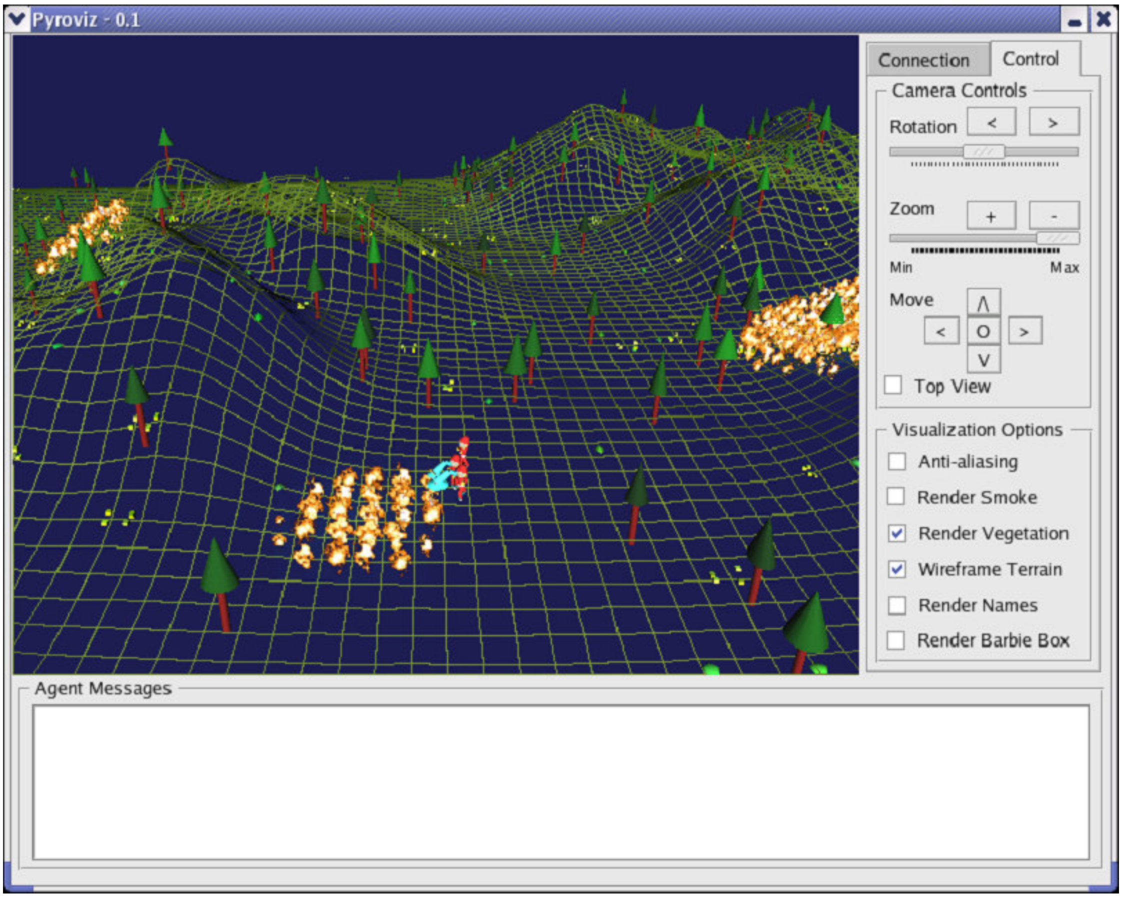

Figure 1.1 The Pyroviz application interface showing the Pyrosim environment and some Agents in action.

1.6. Comparing Motivations

Now that we have presented other possible perspectives on Emotion (as well as our simulation environment) we think it might be interesting to restate our motivations and point out some differences regarding our approach.

First of all, we are not trying to develop Agents that *recognize Human Emotion*. In fact, the Architecture we have in mind does not even consider interaction with Humans: Agents are intended to run autonomously in the simulated environment and the only interaction they may have with the user is through the visualization application or, possibly, through a command-line shell. We admit, nevertheless, that it could be interesting to have mechanisms to recognize Emotion if Human users had deeper intervention in the simulation. For instance, if it were possible for a user to interact with the simulator using a speech recognition system to give Agents high-level command orders, then recognizing Emotion and reasoning about it would possibly be advantageous. However, our work focuses mainly on the development of an Architecture for the *individual* Agent that maximizes its autonomy and adaptability in the environment: interaction with the user is out of the scope of the present work.

We are also not interested in creating *Believable Agents* or Agents that somehow express Emotion. The Architecture we are proposing tries to use Emotional Mechanisms *internally*, for several purposes related with the Agent's performance on the environment. The visualization application we have developed is intended exclusively for helping to understand the behavior of the Agents being developed. The visual representation of the Agent is a secondary concern.

Finally, despite the fact that we are using a very anthropomorphic scenario, we are not interested in modeling or simulating Human behavior. The Agents we are trying to develop could ideally be applied to physical robots intended to operate in real life situations (complex, real-time, multiple goals) Naturally, those robots would have significantly different capabilities from those of Humans. Taking these differences into account, we are aiming to functionally explore an essential feature of Human intelligence, Emotion, and apply it to Synthetic Agents. This is done by studying the role of Emotion in Human intelligence and trying to *transpose* (not simulating) some of its functionalities to synthetic Agents, while making the appropriate adaptations required by different intelligence systems: the Human Brain-Body system vs. a software program running on an electronic computer or a physical robot. Our Agents are not required to behave like Humans would.

1.7. Summary of the remainder of this Dissertation

The remainder of this dissertation will be organized as follows:

- **Chapter 2:** *Human Emotion*. We will make a revision about the most important aspects of Human Emotion especially those concerning the functionalities of Emotion. We will focus on discussing Emotion-Cognition interactions that can be usefully applied or transposed to the development of Agent Architectures. Several authors will be covered. Although this section may be considered rather dense, the concepts discussed therein are essential for subsequent formalizations developed in Chapters 4 and 6.
- **Chapter 3:** *Emotion-based Agent Architectures: State-of-the-Art*. We will make a review of other works that have been developed with similar goals as ours and make some observations about what we consider more interesting and more relevant for our own work's development.
- **Chapter 4:** *Modeling Emotional Mechanisms*. We will introduce the model of Emotional Mechanisms that we propose and explain how it can be integrated in general Agent Architectures. Specifically, we will try to demonstrate how such Emotional Mechanisms can be used to transpose functional Human Emotion-cognition interactions to Synthetic Agents. We will also try to show how this can be advantageous for Agents in general, and especially for resource-bounded Agents that operate in complex and real-time environments and have multiple simultaneous goals to address.

- **Chapter 5:** *The Pyrosim Platform*. We will present in detail the Pyrosim Simulation Platform, and describe each of its three composing elements individually. We will thoroughly describe the simulated environment and discuss its most relevant properties for Agent development. This discussion is important as it will help us clarify some aspects of the Emotion-based Agent Architecture presented in Chapter 6.
- **Chapter 6:** *An Emotion-based Agent Architecture*. We will present the prototype of the Emotion-based Agent Architecture that we have developed, instantiated for the Pyrosim environment. We will present each of the functional modules that compose the Architecture and explain how they are organized in two different processing layers. In complex architectures, such as ours, processing layers and functional modules usually depend on several operational parameters. We will present the parameters associated with each processing layer and functional module and describe the consequences of changing their values. We will also explain the place of Emotional Mechanisms such as “Fear”, “Anxiety” and “Self-Confidence” inside the processing layers and describe their links with the remainder functional modules.
- **Chapter 7:** *The Emotional Mechanisms***.** We will explain how Emotional Mechanisms are used in the Architecture to dynamically influence several of the internal parameters available. We will show how Emotional-like Mechanisms have a deep impact on deciding how processor time is allocated to each of the active Goals and how it is shared among functional modules. We will also explain the influence of Emotional Mechanisms on perception, on planning, on decision-making, on goal selection and also on the Agent’s global attitude towards the environment. We will explain how all these interactions are advantageous for the Agent and how they may even be considered as a requirement for intelligent behavior under resource constraints. We will also explain how Emotional-mechanisms may be changed to alter the global behavior of the Agent, enabling the development of Agents with different “Personalities”.
- **Chapter 8:** *Conclusion*. We will present the summary of the most relevant points of our Emotion-based Architecture and analyze the most important aspects and limitations of Emotional Mechanisms. We will also point out possible lines for future work, namely: including additional Emotional Mechanisms in the current Agent Architecture, exploring Emotional Mechanisms in the Coordination of teams of Agents, and developing the concept of *Personality* for Emotional Agents.

Finally, a listing of all the bibliographic references consulted during this work will be given. The reader can also find some additional information about the mathematical model of Pyrosim in Appendix A.

.

CHAPTER 2. HUMAN EMOTION

2.1. Introduction

In this chapter, we will address several subjects regarding Human Emotion. This will not, of course, be an exhaustive analysis of Human Emotion. We intend to present some of the properties of Human Emotion that have been described in literature and that are especially relevant for developing a functional approach to Emotion.

Throughout this chapter, we will try to generalize the concepts introduced whenever we think it is possible to do so. We will use the word *Agent* if we think that is possible to generalize a given topic about Human Emotion in order to include it in a Synthetic Agent. In some sections we will be using the terms “Agent” and “Human” almost interchangeably.

We will begin by analyzing the context in which Emotions play a functional role: the real world. Next, we will introduce the concept of *Emotional State*. We will then try to establish a connection between Emotion and three related aspects: Goals, Capabilities and Environment. Our notion of Emotional Agent will be introduced and we will explain the process of *Emotional Elicitation*. We will proceed by taking a closer look at Human Emotional phenomena and divide them into three different categories: Specific Emotions, Moods and Emotional Dispositions.

We will then focus on the functional properties of Human Emotion according to what has been described in specialized literature. The aim of this Chapter is then to define those concepts and properties and try to define them in order to make them applicable to Autonomous Software Agents. Finally, we will present this Chapter’s summary and conclusions.

2.2. A Context for Emotion

In order to clarify the reasons why Emotion may have functional properties, let us first analyze a context from where Emotion emerges: Human Agents in the real world. Using the taxonomy provided by [Rus95], the real world may be classified as an environment with the following properties:

- *Inaccessible*: the Agent's sensors cannot detect *all* relevant information for deciding its actions.
- *Non-Deterministic*: although there may be some philosophical debate about the deterministic or non-deterministic nature of the Universe, we consider that from a Human Agent's point of view a real-world environment appears to be frequently non-deterministic.
- *Non-episodic*: the environment does not evolve in well-contained time-slots where we can first perceive and then act.
- *Dynamic*: real-world environment is always changing even if we do not act upon it.
- *Continuous*: in a real-world environment, all parameters range over a continuous set of possibilities.

In [Rus95], the authors also present some examples of environments in accordance with all, or at least with most, of the afore-mentioned properties: "taxi driving", "medical diagnosis system", "refinery controller", or "interactive English tutor". It is also interesting to observe that on the opposite end we would find environments such as "chess", "poker", "backgammon" and "image analysis systems".

On the other hand, Humans are resource-bounded Agents. We have limited intellectual power - including limited memory and inference capabilities - limited perception mechanisms, limited physical capabilities, as well as reasonably severe physiological constraints. At the same time, we have multiple simultaneous goals to be achieved and needs to be satisfied. Besides our *permanent goal* of maintaining our biological health, numerous personal (physical and intellectual) and social goals emerge from our interaction with the world and need to be attained. With such an apparent mismatch between our many goals and our limited capabilities, it seems quite reasonable to admit that Human Agents should have little chances of being successful in the real world (for an interesting and deeper approach on this subject refer to [Mof00]).

2.3. Emotional State

Before proceeding with the discussion on Human Emotion, we shall make some previous assumptions for the sake of generalization. We now introduce, at this point, the notion of *Emotional State,* fundamental for the subsequent sections.

Let us first assume that an Agent operating in a complex environment requires a set of specific capabilities to deal effectively with such complexity. These capabilities allow the Agent to endure in the environment and address multiple other needs. Let us also assume that if such capabilities, although limited, are reasonably sophisticated (either as a result of a long process of evolution or as a consequence of thorough design and implementation), they are almost certainly related to various operating parameters that control their individual activity.

The results achieved by the Agent in the environment will depend on how the specific parameters of its individual Capabilities are instantiated. If several of such capabilities exist, intended to address multiple needs, the Agent will be required to achieve a global state where all operating parameters of all available Capabilities are optimally instantiated.

Definition 1:

An Agent's *Emotional State* is a particular instantiation of operating parameters for the complete set of Capabilities (physical or cognitive) it possesses. An Emotional State defines a global operating context in which Agent's available resources are concentrated in the achievement of a specific Goal or set of Goals.

We will now proceed with a more specific discussion on Human Emotion where we will try to introduce some additional concepts to reinforce the functional perspective of Emotion.

2.4. Emotion and Situated Agents

The work developed by Damásio [Dam94] established a strong relationship between the Human ability to deal with real world situations and Emotions. Damásio undertook several studies in his patients that had suffered damages in specific brain structures (pre-frontal cortex). As a result from such damages, these patients had developed major changes in their emotional behavior and were unable to respond to emotionally rich stimuli, such as, for instance, violent or sexual content images. Despite showing average or above average results in regular laboratory IQ tests, these patients were very unsuccessful when confronted with situations that required them to make real-world decisions, either at a personal, social, or professional level.

Damásio concluded that Emotion should play a decisive role in allowing humans to deal with the complexity of the real world. Emotions[3] would *speed up* Cognitive processes, creating various kinds of *shortcuts* for deliberation in a given *situation*. Without such shortcuts people seem to be unable to deal with the complexity of the real world with the usual time constraints: Emotions provide mediation mechanisms between cognitive capabilities and the environment. Interestingly, the notion of Emotions as a *mediation* mechanism between a Situated Agent and its environment had already been pointed out by Fridja:

Many emotions can be defined by an intentional structure: that of maintaining or changing a given kind of situation, and to doing that in certain ways, with respect to certain aspects

[3] At this stage, we will not be providing a formal description of Emotion. In fact, throughout this thesis we will avoid giving an exact definition of what an Emotion is because a consensual definition is yet to be found in literature. However, in Chapter 4 we will try to more carefully formalize some of the concepts related to Emotion.

of the given situation and with regard to given kinds of objects in those situations. [Frijda, The Emotions, p. 98]

This perspective can be better understood by calling attention to the simultaneous link that Emotion establishes with the perception the Agent has of the environment and simultaneously of its own capabilities to deal with it. As mentioned in [Mof00], Emotions are connected to two essential issues. Firstly, Emotions are related to the identification of internal or external events/situations that are relevant to specific Agent's Goals (or *concerns*[Fri86]). Such situations may be considered either positive (in the case of opportunities) or negative (in the case of threats), regarding the Agent's chances of achieving those Goals. Additionally, Emotions are simultaneously related to the assessment of current capabilities and resources (or *coping potential* [Fri86]) that the Agent has or can make available to deal with the identified situation.

According to the authors, a given Emotion will emerge depending on the relation established between the two previous parameters. For example, the Fear Emotion would result from the identification of a situation or event that is particularly dangerous to the Agent (i.e. to a Goal) and, simultaneously, from the fact that the Agent has limited capability (i.e.: physical or computational resources.) to respond appropriately to that situation. The emergence of Emotions according to certain conditions is a process known in literature as *Emotional Elicitation*, which we will address in the next section.

We will now make our first generalization regarding Human Emotion based on the previous definition of Emotional State.

Generalization 1:

An Agent reaches a given Emotional State depending on its perception of how favorable the environment is for the achievement of its Goals, considering its own current resources. The Emotional State generates a global operating context appropriate for the achievement of a given set of Goals and that, at same time, optimizes resource allocation for that purpose.

Based on this generalization we may now introduce another fundamental definition:

Definition 2:

An Emotional Agent is an Agent that is able to operate under different Emotional States. An Emotional Agent will always be operating under *one* specific Emotional State.

The implications of such definition will become clear in the next sections. Nevertheless, it is possible to say that, according to this definition, Humans may be considered Emotional Agents. Let us now take a closer look at the process of Emotional Elicitation.

2.5. The Process of Emotional Elicitation

Emotional Elicitation is a matter that generates great controversy among researchers [Ekm94, Pic97]. The main issue here regards the basic nature of the elicitation process. Some researchers, especially those coming from the field of psychology, advocate that Emotional Elicitation itself requires some form of high-level cognitive processing of the corresponding antecedents (appraisal theories [Fri86, Ort88]). On the other hand, researchers coming from the field of neurobiology [Led96] defend that some emotions can arise from physiological or lower level stimuli that do not involve any high-level cognitive circuitry (neocortex or hippocampus).

However, there is much less dispute around this issue if a wider concept of "form of cognition" is considered, one that includes both high-level cognitive processes and also basic sensory information processing. From this perspective, it is possible to gather the two contending views and reach a comfortable compromise. We will also adhere to this broader concept of "form of cognition", believing that it is possible to devise a general functional model of emotional mechanisms irrespective of the nature (deliberative or physiological) of the *Emotional Elicitation* process.

As we previously mentioned, Emotions are deeply related to the evaluation of the Agent's own capabilities to achieve a given goal. Goal, in this context, includes not only explicit formulations of desirable future states that the Agent seeks to achieve, but also implicit states that *must* be reached or maintained. Emotions reflect the Agent's capability (knowledge, action set, physical robustness, etc.) to cope with the current state of the environment when trying to achieve one or more of its goals. This concept is known as *coping potential* [Fri86] and is essential to a functional view of Emotions. For instance, *Frustration* may reflect a systematic inability to deal with the environment when attempting to achieve a certain goal. *Fear* may reflect the fact that the current environment situation is highly unfavorable to the accomplishment of at least one *goal* (probably a very important or even vital one), and that the Agent does not have the necessary capabilities to cope with the situation. It is interesting to note that *Fear* is considered to have a very strong neurobiological / sensory dependency (the fear generated by the approaching of a fast object or by a high volume sound) but it also has a highly cognitive dependency (the *Fear* of a major inversion in the market share). Nevertheless, we are obviously dealing with "*Fears*" of very different nature.

Therefore, and regardless of the nature of the eliciting process (cognitive or neurological), Emotion Elicitation involves the *evaluation* of the chances of achieving a given *goal*, taking into account both the state of the *environment* and the internal state of the *Agent* itself. *Emotional Elicitation* can be abstractly seen as a process based on an *evaluation function* involving:

- a goal
- the Agent's capabilities
- the environment state or an event.

The next figure illustrates these relations:

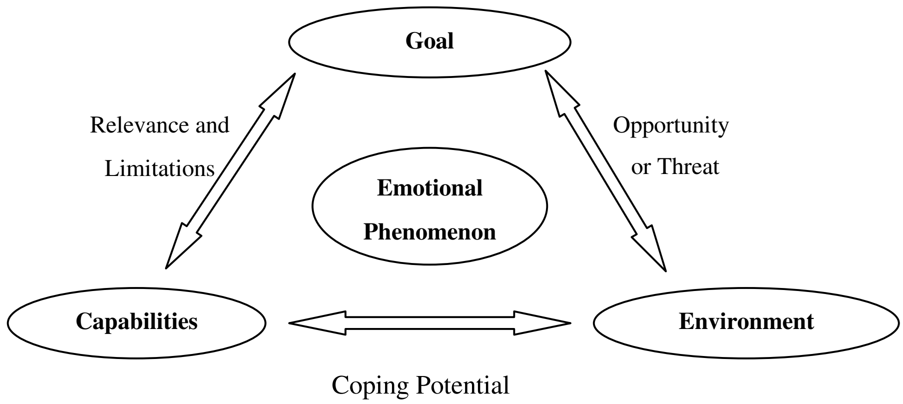

Figure 2.1.Emotions may be seen as the center of a triangle formed by (i) Agent's Goals, (ii) the State of the Environment and (iii) the Agent's Capabilities to ensure its Goals under the current Environment.

We may now introduce an additional definition that generalizes this concept in order to make it applicable to Synthetic Agents:

Definition 3:

The *Emotional Elicitation* is a process that promotes a given Emotional State after an evaluation based on an *Emotional Evaluation Function (EEF)*. The evaluation depends on three parameters: the Agent's set of Goals <G>, the state of the Environment <Es> and the Agent's Internal State <Is>. If <E> denotes the change in the Agent's current Emotional State, then:

<E> = EEF(<G>,<Es>,<Is>)

We will be developing this issue further in Chapter 4. For now, it suffices to say that such Emotional Evaluation Function may possibly be very simple (for example a comparison function), and the same may be said about the data contained in both the Internal State <Is> and the State of the Environment <Es>, i.e. direct sensory information instead of complex Belief-like structures.

2.6. A Simple Taxonomy of Emotional Phenomena

Up to this moment, we have been referring to Emotion without paying attention to the fact that Emotion is a complex and broad concept and that there are many phenomena that we are used to indistinctly call Emotion. Emotions can be subdivided in, at least, three different categories of Emotional phenomena, namely:

1. *Specific Emotions;*
2. *Moods*;

3. *Emotional Dispositions*.

Divisions can be made according to the following criteria:

- **Object/Antecedent:** cause or pre-conditions that trigger the emotional phenomena;
- **Intensity:** how strong is the influence of the emotion on the Agent;
- **Duration:** time span that includes the rise and fall of the emotional phenomena;
- **Consciousness:** whether the Agent is conscious or not about the occurrence of the emotional phenomenon.

It is important to make this division because, although all these phenomena share a common functional logic and high-level model, they have distinct impacts on cognition. Thus, for now, we shall try to differentiate them as clearly as possible, without alluding to their functional role.

Specific Emotions

Specific Emotions are, by definition, emotional phenomena that can be quite clearly differentiated. Although cognitive researchers are still debating which emotions exactly constitute the set of Specific Emotions [Ekm94], *Anger*, *Fear*, *Joy*, *Surprise*, *Disgust* are usually considered part of that set. Specific Emotions, as those we have just referred, have well-defined objects and are usually very intense. However, they usually occur over a reduced time span (few seconds or minutes). Because of their strong intensity, an Agent is, most of the times, clearly aware of its existence.

Moods

On the other hand, *moods* encompass a different set of emotional states such as *Pleasant*/*Unpleasant*, *Anxious*/*Relaxed*. Contrary to Specific Emotions, moods may have no clearly defined object or antecedent. In addition, because they are usually less intense, they may remain unconscious to the Agent for most of the time. Moods may be originated by an intense or recurrent occurrence of a Specific Emotion (e.g.: *Anxiety* after several fearful episodes) or simply by environmental factors (e.g.: weather). They may last for hours or even for a few days. We will also include in this category states such as *Self-Confidence* and *Frustration*, a choice that will be made clear shortly.

Emotional Dispositions

Emotional Dispositions (sometimes called *temperament*) represent yet another step regarding duration of emotional phenomena. *Chronic Anxiety* and *Depression* are examples of Emotional Dispositions. They may last for several months or years but tend to be inactive most of the time. They are assumed to be partially influenced by genetics and may appear early in life. Emotional Dispositions have also been observed to arise in patients who have been prescribed certain medication or have been subjected to neuro-surgery [Dam94].

Table 2.1 Summary of Human Emotional Phenomena

Emotional Phenomena	Elicitation Time	Duration	Antecedent	Intensity
Specific Emotions	Short	Short	Identifiable event or condition usually limited in time	Strong
Moods	Medium	Medium	Environmental or recurrence of Specific Emotions	Strong or Medium
Emotional Dispositions	Long	Large	Usually environmental or intrinsic. Difficult to identify	Medium

Some of the previous Emotional phenomena may occur combined: each Emotional phenomenon makes its contribution to the global Emotional State of the Agent. However, during a given time period, the influence of one single Emotional phenomenon usually prevails over the others, becoming the dominant Emotional phenomenon and, therefore, the major component of the Emotional State. This happens to create the internal context needed for the Agent to achieve the most relevant Goal at that moment. After some time, the dominant Emotional phenomenon will fade (or not), allowing other Emotional phenomenon, related to another Goal, to become dominant.

In the next section, we will analyze some of the conditions that interact with these dominance changes.

Some issues regarding the Dynamics of Emotional Mechanisms

The typical activity period of emotional phenomena varies for each of the three categories, ranging from a few seconds (Specific Emotions) to, eventually, many years (Emotional Dispositions). Additionally, for each of those categories the elicitation process duration also has a typical value. For instance, a Specific Emotion such as *Angriness* may be elicited in just a few seconds (we all know that). On the other hand, Depression, which can be considered either a Mood or an Emotional Disposition, has usually much longer elicitation periods. Both the activity and duration period of emotional phenomena can be understood according to two functional requirements: *urgency* and *temporal consistency.*

Fast elicitation times are *functionally* appropriate when the eliciting condition is an event/situation that demands an *urgent response* from the Agent. Note that this does not necessarily imply a reactive response from the Agent. It simply creates internal conditions, i.e. an Emotional State, to promote the appropriate answer (may it be braking a car after detecting an approaching object or quickly constructing the appropriate verbal answer to an insulting comment). On the other hand, long elicitation times are compatible with the need to keep up with slow environment drifts or with recurrent events that do not pose an urgent demand on the Agent but may have great influence in the long run. As for the activity period of emotional phenomena, shorter activity periods seem appropriate when the corresponding Emotional State is not required to last, or may not be sustainable for long periods. They are also usually related with specific events or conditions that can be clearly identified in time. On the other hand, longer activity periods seem useful when there is a need to cope with situations, possibly internal ones, which may be difficult to specifically identify or whose duration may be undefined. In these cases, the Agent should keep a *consistent response* over a longer (probably undefined) time.

2.7. Functional Properties of Emotional Phenomena

In the previous section we have briefly alluded to a functional property of Emotional phenomena, that of creating an appropriate operational context for the Agent. In this section we will try to analyze this and other different functionalities and formulate them in a more general way, so that they can be transposed to Synthetic Agents.

2.7.1. Emotion as Information

Real-life environments are usually inaccessible, non-episodic, dynamic and continuous. People deal with this complexity everyday. However, when someone is asked to rate the overall life satisfaction, one will not trigger a multi-criteria evaluation process to reach a conclusion. This task would certainly involve a great amount of factors, some of them uncertain or difficult to identify and quantify. Instead, people will find the answer simply by asking themselves "How do I *feel* about my life?" [Sch00].

This behavior, which may also be adopted in more specific questions, is based on the premise that our Emotional phenomena seem to be capable of *condensing* large amounts of dispersed information into a single and easily identifiable information unit, the *Intensity of Emotional-phenomena*.

In complex environments, information may be difficult to obtain because sources are disperse and noisy. Alternatively, the amount of information available may be so large that Agents will simply not have enough computational resources to process it. Therefore, the information *condensation* provided by the feeling of Emotion can actually become a very functional property. The intensity of Emotional phenomena can be used as supplementary or alternative input to certain high-level cognitive processes, such as decision-making or learning, which would otherwise be impossible to compute or very unreliable, if based exclusively on external information.

Since Emotions are essentially linked to Agents' Goals and Capabilities, this *condensation* works as a very functional *relevance filter*. Emotional phenomena are capable of amplifying the most relevant data for the Agent (external *and* internal), taking into account the Agent's current capabilities to deal with those events.

The information condensed in the Intensity of a given Emotional phenomenon can be useful in complementing the information collected by the perception system. The extent to which the information conveyed in Emotional phenomena is actually used by other cognitive processes depends on several factors. In [Bow00], the authors present the *Affect Infusion Model* (AIM) that describes the conditions in which Emotional[4] information seems to have more impact on cognitive processes. This issue is related to the topics of the next section, so we will come back to it later.

2.7.2. Processing Strategies

Besides providing alternative and complementary sources of information, Emotional Phenomena also have a deep impact on the way our cognitive machine processes all available information [Bow00]. Four different information-processing strategies have been identified and are optimized to address four different Agent-Environment situations:

1. *Direct Access*. This processing strategy relies on the use of fixed pre-existing structures/knowledge. Rule-based mechanisms and case-based reasoning fit in this category. It is the simplest strategy and corresponds to a minimization of computational effort.
2. *Motivated Processing*. This strategy is employed when the Agent has a defined set of goals that direct (motivate) all information processing. Examples are: searching knowledge for a suitable answer or producing plans of actions and executing them until achieving a given goal. Motivated Processing is usually computationally intensive.
3. *Heuristic Processing*. It is a processing strategy used when there is no specific, clearly defined goal or whenever there is insufficient information or cognitive capacity available to achieve a goal. *Heuristic Processing* is also used under totally new situations where it is not possible to rely on pre-existing structures. It is also a minimum effort strategy.
4. *Substantive Processing*. This is the most complex information processing strategy that is employed in complex information structures or in atypical tasks. *Substantive Processing* is used, for instance, in learning, in interpreting and in performing judgments or associations.

These four processing strategies have different requirements regarding computational needs. *Direct Access* and *Heuristic Processing Strategies* require reduced computational resources. On the other hand, both *Motivated* and *Substantive Processing Strategies* are much more demanding, in spite of probably leading to better or safer results. It is possible that for some tasks (e.g. choosing the appropriate behavior for a situation) more than one processing strategy will lead to satisfactory results

[4] We diverge from the original description of the authors by using the terms *Emotion* instead of *Affect*.

(e.g.: *Heuristic Processing* or *Substantive Processing*). This is an important possibility in a resource-bounded Agent because, depending on the computational resources the Agent has available at the moment, and also on the urgency of the situation, one of the possible strategies will eventually be more appropriate. If it is possible to choose between two or more strategies without jeopardizing the accomplishment of the task, then, the most economic strategy should be preferred. The processing power saved may then be made available to other concurrent processes.

Experimental results suggest that the choice of a particular processing strategy depends strongly on the Intensity of Emotional phenomena. Positive valance Emotional phenomena, such as happiness and self-confidence, favor the choice of simplified or relaxed processing strategies [Sch00] with top-down and high-level approaches. This behavior seems compatible with the adoption of the previously described *Direct Access* or *Heuristic Processing Strategies*. On the other hand, people under the influence of certain negative Emotional Phenomena, such as sadness, seem to adopt more systematic processing styles, showing great care for details and following bottom-up approaches. Again, this behavior seems to be similar to the description of *Motivated* or *Substantive Processing*.

The correlation between Emotions and processing strategies provides support to another functional property of Emotion [Oli03]. If, in fact, negative Emotions should emerge whenever the Agent is not being successful in achieving its goals, it seems optimal that they should be able to promote:

1. the adoption of processing strategies that focus on specific goals, such as *Motivated Processing Strategies*, or
2. the adoption of safer strategies that are capable of dealing with more complex information and allow a global increase in the Agent's performance, such as *Substantive Processing Strategies*.

Although these strategies require a higher computational effort, it is reasonable to assume that their adoption has a good chance of compensating.

In contrast, positive Emotional Phenomena should emerge in situations where the Agent is having success in coping with the environment and in achieving its own goals. In such situations, it may be optimal to adopt less expensive, more relaxed processing strategies and simply employ structures and knowledge that have proven to be effective before. This would be accomplished by adopting *Direct Access* and *Heuristic Processing Strategies*, resulting in a lower consumption of computational resources. The released computational resources could be allocated in non-real time processes (e.g.: data-mining, pattern detection, inference of rules) that may provide long-term adaptability to the environment.

If, as a result of adopting *Direct Access* or *Heuristic Processing Strategies,* the Agent decreases its performance level and endangers some of its goals, negative Emotions (e.g.: sadness, fear, frustration) should arise, promoting the adoption of *Motivated* or *Substantive Processing Strategies* that will have a corrective effect.

In our opinion, this resource management mechanism is another of the functional properties of Emotion, which seems to be very useful in highly parallel Architectures with multiple concurrent processes, as is the case in Humans.

Finally, and resuming the topic of the last section, it has also been identified that Humans tend to give more value to information conveyed by Emotional Phenomena when operating under certain Processing Strategies. Substantive Processing and Heuristic Processing make considerable use of Emotional information, whereas Motivated Processing and Direct Access strategies are relatively immune to its influence. Having the conditions that lead to the adoption of each processing strategy in mind, it is interesting to note that both Substantive Processing and Heuristic Processing are associated with difficult situations for the Agent, either because the environment is intrinsically complex or because the Agent's capabilities are assumedly not enough. Using Emotional information seems to provide a way to compensate for these limitations. Once more, we believe that the functional role of Emotion is evident.

In the next figure, we plot each possible Processing Strategy in a 2D dimensional map. The axes refer to the amount of computational effort required for computation and the influence of Emotional Information on global information processing.

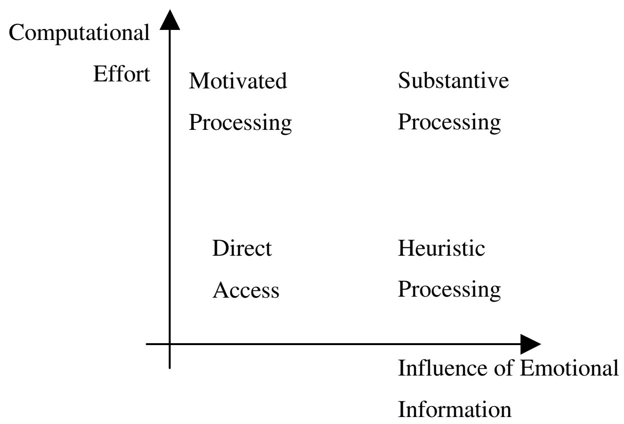

Figure 2.2 The four different processing strategies compared, regarding the Computational Effort required for their execution and the corresponding influence of Emotional Information.

2.7.3. Process Control and Operating Modes

In the two previous sections, we described functional properties of Emotion that globally interact with the Agent's Cognitive machinery. We will now try to analyze more localized interactions, in particular those concerned with specific cognitive processes or capabilities.

First, we must observe that most of our cognitive functions have several possible *Operating Modes* available. Cognitive functions may be performed with a great degree of flexibility that results from our need to cope with complex and dynamic environments. For instance, planning our day involves making many predictions and decisions based on some more or less uncertain decisions. We may obtain several possible plans with varying degrees of detail, depending on the effort and time you invest in planning. Adopting either a pessimistic or optimistic attitude towards your day will have impact on the chances of success of such plan. You may consider a large set of possible options for your day or, instead, focus on a reduced set of possibilities. Additionally, you may even plan your day thinking on what you will need to do on the following days. There are many possible variations in a planning task that seem especially compatible with specific contexts (e.g. regular day vs. the imminent conclusion of an important project vs. the approaching of a long trip abroad). With the appropriate adaptation, the same analysis can be made regarding the Operative Modes of other cognitive functions or tasks. For instance, similar operating flexibility could be considered in a task such as deciding the best move or strategy in a chess game.

Clearly, different Operating Modes require different computational resources. Usually, tuning processes for best-quality results implies increasing computational effort and response time. On resource-bounded Agents operating in real-time environments, the appropriate balance between performance quality, resource used and time must be efficiently attained (in [Oli03] we have addressed the compromises involved in the case of Planning).

Following our previous argumentation, it seems natural that Emotional Phenomena should somehow be involved in regulating the Operating Mode of specific cognitive functions. For example, an Emotional State such as *Fear* should promote the reduction of processing time, because it is activated by an urgent condition in the environment that needs to be promptly dealt with. *Fear* should also be able to bias the set of possible results in some direction, for example by changing the heuristics employed. On the other hand, an Emotional State such as *Anxiety*, which is usually activated when difficult forthcoming situations are detected, should promote an increase in the results' level of detail and depth. The corresponding increase in the computational effort would be justified by the need to maximize global success chances in dealing with the problematic situation. In both cases the balance thus obtained seems "natural" and functional. The next figure shows a high-level illustration of the interactions between Emotional Mechanisms and specific Cognitive Processes.

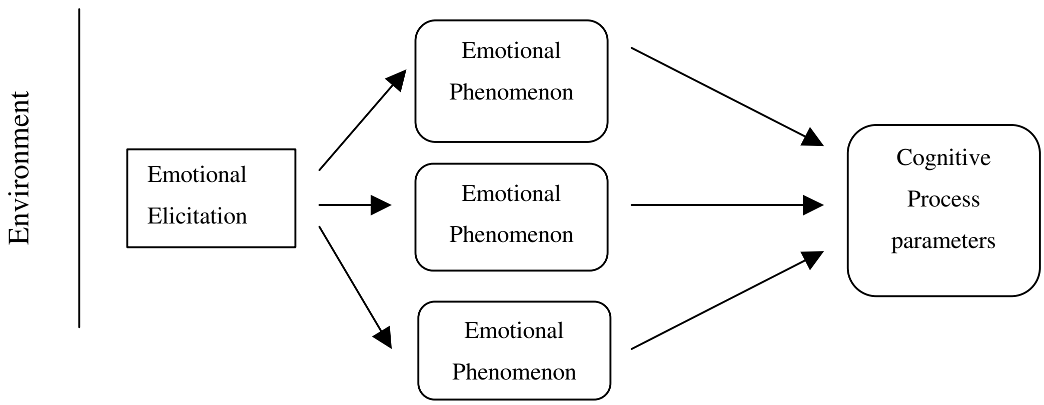

Figure 2.3. The influence of Emotional Phenomena over the parameters that control Cognitive Processes.

It is worth noting that connecting Emotional Mechanisms and Operating Modes is not equivalent to using the Emotional State to determine the result of a given process. The Emotional State alone does not usually determine what the actual result of a cognitive function will be. We do not assume the existence of a *direct* connection between the Emotional States and the result of a cognitive function: Emotional Mechanisms are not intended to be "reactive". Instead, what an Emotional State should be able to do is modulate the *process* that leads to a certain result.

2.8. Summary and Conclusions

In this Chapter, we analyzed some basic concepts regarding Human Emotion, focusing on possible functional interactions with Cognition. We tried to generalize the concepts of Human Emotion in order to make them applicable in the more generic context of Autonomous Agents. We showed how Emotion should be placed in the middle of a triangle made by (i) Agent Goals, (ii) Agent Capabilities and (iii) environment situations or events. We have described the process of Emotional Elicitation and a Taxonomy of Emotional Phenomena that differentiates three distinct Emotional Phenomena: Specific Emotions, Moods and Emotional Dispositions.

We introduced the following concepts:

- *Emotional State*
- *Emotional Agent*
- *Emotional Evaluation Function*
- *Processing Strategies*
- *Operating Mode*

Finally, we pointed out some functional Emotion-Cognition interactions namely:

1. *Emotion as Information or Emotional Information*: The Intensity of Emotional Phenomena may be used as input for specific cognitive processing, compensating the unavailability or impossibility of dealing with standard sensory information;
2. *Emotion as a control mechanism for Processing Strategies*: Emotional States promote the adoption of specific patterns of information processing, as well as different levels of Emotional Information usage. Controlling processing strategy results as a global coordination mechanism and as a global resource manager.
3. *Emotion as a mechanism for Operating Mode configuration*: Parameters associated with the operation of specific Cognitive Processes may be set according to the Emotional State, thus creating the appropriate operating context to achieve a given goal under certain constraints (urgency, available resources).

CHAPTER 3. EMOTION-BASED AGENT ARCHITECTURES: STATE-OF-THE-ART

3.1. Introduction

In this chapter, we will present a brief analysis of the work developed by other researchers that have deeply influenced our work. Although the amount of research in this area has increased over the last few years and numerous authors now address Emotion from a functional point of view, we will focus on three particular researchers whose work is considered a reference:

1. Aaron Sloman for his broad view regarding Emotion, Cognition and Agent Architectures;
2. Juan Velasquez for his practical and modular Architecture and for the interesting hardware implementation of an Emotion-based Agent; and
3. Lola Cañamero because of her clear view about the functional role of Emotion in Agent Architectures.

We will try to describe the work of these three with some detail because several of the concepts we developed emerged from their research. Many other researchers have influenced our work, but describing all these influences would be very difficult and confusing. Therefore, and despite some unfairness in the choice, in this Chapter we will not make a review of the work of other authors such as Beaudoin [Bea94], Botelho and Coelho[Bot95] [Bot01], Custódio et al.[Cus99], Wright[Wri97] and Ventura[Ven00]. We will only make an exception to briefly describe the work developed by Gadanho as it follows a very original view regarding the use of Emotion in the learning context.

Finally, we will end this Chapter by making a review of the most relevant points, in our opinion, of the works developed by these researchers.

3.2. Aaron Sloman

Aaron Sloman has been one of the most active and influential AI researchers in the study of Emotions and related concepts [Slo81][Slo98][Slo00][Slo01]. Sloman is the lead researcher of the

Cognition and Affect Group (CogAff) located at Birmingham University. The CogAff group has been dedicated to the study of Agent Architectures for years and has had the involvement of several researchers with relevant work on Emotion in AI (notably Luc Beaudoin[Bea94] and Ian Wright[Wri97]). The work and ideas developed by Sloman and other members of the CogAff Group founded what is known as the "Birmingham School" in respect to the relation established between Cognition and Emotion.

Sloman has addressed the study of Intelligent Agent Architectures using the "Design-based Approach" [Slo00][Slo01]. Instead of trying to model particular Agent Architectures exclusively based on what is known about physiological structures and neurological processes involved in intelligent behavior, the "Design-based Approach" consists in exploring the space of all relevant Architectures that may explain intelligent behavior and Agent control. According to Sloman, this exploration will help us understand which Architectures are able to explain a specific form of intelligence, after being properly instantiated. In fact, Sloman seeks to develop Architectures that are capable not only of explaining Intelligence, both in Humans and in insects, but also Architectures that may effectively explain other observable phenomena, for example, in patients with brain diseases.

3.2.1. The Architecture

The Agent Architecture proposed by Sloman involves the combination of two other traditional Architectures: the "three towers" model and the "three layers" model [Slo01].

The "three towers" model comprises three parallel subsystems that are conceptually vertical within the Agent Architecture:

1. the Perception Subsystem that is responsible for extracting data from the environment where the Agent is operating;
2. the Action Subsystem that allows the Agent to act upon the environment;
3. the Central Processing subsystem that mediates perception and action and is capable of controlling both of these subsystems.

In order to achieve a globally successful performance, there is usually a great interaction level among these three subsystems. For example, the Action Subsystem may provide direct feedback to the Central Subsystem (proprio-perception) or it may interact with the Perception Subsystem to allow effective coordination of some tasks (e.g.: hand-eye coordination). Moreover, despite the existence of the three conceptually different subsystems, the actual implementation of an Architecture using the "three towers" model may blur such well-defined distinctions. Some of the Architecture's components (e.g.: visual system) may belong both to the Perception Subsystem and to the Central Subsystem.

The "three layers" model is based on the horizontal division of the Architecture in three layers, each of them capable of providing different forms of information processing and control functions. This model is deeply inspired in Evolutionary theories: each of the layers would have been developed in different stages of the evolutionary path of organisms based on the capabilities of the previous layer. The "three layers" model consists of the following layers:

1. Reactive Layer. This layer includes reactive mechanisms, capable of delivering automatic responses when triggered by specific environment (internal and external) conditions. The reactive layer contains mechanisms capable of implementing pattern matching functions, condition-action rules and other direct input-output functions. They are also responsible for providing a basic set of Motivations. Mechanisms such as these may achieve great complexity levels and are usually implemented with great degree of parallelism (e.g.: neural nets).
2. Deliberative Layer. This layer would have resulted from the evolution of some of the mechanisms belonging to the Reactive Layer. The most important evolution in this layer is the development of reasoning capabilities about past, present and future events (what if reasoning). This layer also supersedes the previous one by adding more sophisticated memory systems and symbolic reasoning capabilities. In spite of these extra elements, the Deliberative Layer has limited processing resources.
3. Meta-Management Layer. This third layer includes mechanisms intended to monitor, evaluate and redirect processes executed in the Deliberative and Reactive Layers. This layer would have been developed in order to provide efficient control of all the machinery operating in the two lower layers. The mechanisms implemented at this layer may be reactive or deliberative. Processing resources at this level are also limited.

The Architecture presented by Sloman, superimposes the "three towers" model and the "three layers model" forming a hybrid Architecture with nine distinct conceptual components (3x3 grid). The processes running in each of the three layers operate concurrently and deal with Perception, Central Processing and Action at different levels of abstraction.

3.2.2. Motives, Global Alarm Mechanisms and Variable Attention Filter

Sloman observes that Agents operating in complex and real-time environments are usually compelled by several Motives (similar to Fridja "multiple concerns"). Some of these Motives will be active simultaneously and can be considered competitors regarding the Agent's available processing resources. Conversely, other Motives will not be permanently active, requiring special motive-generator mechanisms to be activated when certain conditions are found. Since the Agent has limited resource capabilities, some information processing mechanisms (in particular those located in the

Architecture's upper layers) may not be able to respond fast enough to specific conditions occurring in real-time environments (either dangers or opportunities). Moreover, urgent Motives may require processing resources not available at that moment, possibly because they are being used to process less urgent Motives.

Sloman tackles the problem of resource limitation by including in his Architecture "Global Alarm Mechanisms". These Global Alarm Mechanisms receive information from every component in the system and, by using fast pattern matching procedures, they are able to detect situations whose urgency requires a change in the Agent's internal processing strategy. When such a situation is detected, Global Alarm Mechanisms immediately send system-wide interrupt signals in order to stop current processes and trigger the whole system's redirection to conveniently deal with the urgent Motive or situation. This may involve switching the Motive that is receiving processing resources or it may even require stopping computationally heavy processes, followed by the activation of faster reactive mechanisms that can provide a more efficient response to the urgent situation. However, Sloman does not clearly explain how these Global Alarm Mechanisms should be implemented. It is not clear if they are exclusively located in the reactive layer, or if they may be eventually located in higher layers.

Global Alarm Mechanisms can help the Agent achieve a balanced use of its capabilities. However, they may also become very unproductive, if the conclusion of deliberative and meta-management processes is systematically delayed or postponed as a result of frequent interruptions. In order to reduce this possible negative effect of Global Alarm Mechanisms, Sloman has devised the Variable Attention Filter mechanism, which is intended to filter some of the interruptions targeted at specific processes by raising the minimum urgency level that may interrupt the process. This will result in what can be considered as a state of "concentration".

Therefore, when Agents need to execute tasks that require continuous deliberative or meta-management processing, the Variable Attention Filter will increase the minimum urgency threshold to ensure that such process will not be interrupted, unless the signal is generated in response to a very urgent condition or Motive.

3.2.3. Emotion

Sloman draws a relationship between Emotion and the interactions established between the Architecture's subsystems. In fact, Sloman argues that Emotional States arise naturally from interactions established between these subsystems, with no need for a dedicated Emotion-generating mechanism. Sloman divides Emotional Phenomena into three categories directly connected to the three layers in the Architecture. Following the definitions proposed by Damásio [Dam94] closely, Sloman agrees on the existence of Primary and Secondary Emotions but introduces an additional concept: Tertiary Emotions.

According to Sloman, Primary and Secondary Emotions result from the interactions established between Alarm Mechanisms and other subsystems located in the Reactive and Deliberative Layers [Slo98]. Primary Emotions, such as being startled, frozen with terror or sexually aroused [Slo01], are supported by Alarm Mechanisms located in the Reactive Layer, which are mainly concerned with processing sensory information (inside and outside the environment) and triggering fast Reactive Mechanisms. This interaction is believed to be similar to the role of the Limbic System in Humans.

Secondary Emotions are supported by mechanisms in the Deliberative Layer and include emotions such as apprehension, relief and other semantically rich emotions that require deliberative capabilities. Secondary Emotions result mainly from Alarm Mechanisms concerned with evaluating internal cognitive responses (e.g.: the chances of success of a risky plan) that are not directly linked to the perceived environment.

Tertiary Emotions, on the other hand, result from the mechanisms located in the Meta-Management Layer. Therefore, Sloman argues that they are probably exclusive to Humans. They include emotions related with thought and attention control such as infatuation, humiliation and thrilled anticipation. These Emotions interfere with Deliberative processes by diverting the attention from current tasks and triggering introspective processes, in spite of the Agent's attempt to ignore such interruptions.

3.2.4. Overview and Comments

It is important to note that the perspective followed by Sloman regarding Emotions is focused on explaining their occurrence as a result of Architectural requirements. Sloman's works focus mainly on Agent Architectures. For the author, the concepts regarding Emotion are a consequence of the evolution of such Architecture. In spite of this, Sloman's view on Emotion is very broad and the related concepts are supported by solid and coherent arguments.

Sloman agrees with the general opinion that Emotional phenomena are mechanisms whose existence is inherently associated with resource-bounded Agents that need to operate in real-time environments. However, in [Slo00] Sloman disagrees with the point of view shared by several other researchers (notably Damásio), by stating that Emotions should not be considered an absolute requirement for Intelligence. In fact, Sloman argues that Emotions are simply side effects of the mechanisms needed to overcome an Agent's resource limitation.

The key concepts around Emotion are Alarm Mechanisms and Interruptions. Emotions are mainly seen as effects of existing control mechanisms intended to:

- detect situations or motives that need urgent response from the Agent;
- trigger the appropriate redirection of processing resources at different levels of abstraction.

In this way, Sloman seems to prefer a hard resource scheduling policy (interrupt and swap) instead of an alternative softer and finer-grain control of Agent's processing resources.

Sloman's views about Emotions come from a more general perspective on cognition that we also share. However, Sloman is much more oriented towards a global descriptive theory of cognition and less focused on the specific functional properties of Emotion. Nevertheless, some of the underlying concepts behind Sloman's theories are close to the ones we presented in the last Chapter, namely those regarding environment evaluation and resource control in complex architectures.

3.3. Velásquez

Juan Velásquez' work [Vel98a] [Vel98b] represents a very interesting approach to the study of Emotional Mechanisms because it includes a simple, modular and extendable Architecture that explicitly supports Primary and Secondary Emotions, as defined by Damásio [Dam94]. Additionally, Velásquez has also developed a physical robot whose decision-making process is based on the proposed Architecture, providing an alternative hardware implementation of Emotional Mechanisms.

3.3.1. The Architecture

As discussed in [Vel98b], Velásquez follows a Biological perspective concerning the study of Emotions. The author considers Emotions as biological phenomena that have been conserved through various stages of evolution and that have a deep relation with survival and adaptation. Velásquez establishes as a basic assumption for the development of computational Emotional Mechanisms the need to understand and model the neurological structures that support Emotion. This goes in the opposite direction of other more descriptive approaches as those leading to Cognitive Appraisal Models. In this sense, Velásquez' approach is quite different from many researchers', who have based their Architectures exclusively on Cognitive Appraisal Models (such as the OCC Model [Ort88]), since it includes both Cognitive and non-Cognitive components of Emotion.

Velásquez also stresses the importance of differentiating Emotions (i.e. Specific Emotions) from other Affective Phenomena (i.e. Emotional Phenomena) such as Moods or Temperament. Additionally, Velásquez argues that while developing Emotional Mechanisms the following conditions should be observed:

- Emotional Mechanisms should be modeled within a broader Architecture that integrates Perception, Motivation, Behavior, Motor-Control, etc.
- Development should be incremental and Architectures should be modular enough to accommodate new properties and functionalities as researchers develop deeper insight into Emotion.

- At each step of development, the Architecture should be able to generate complete Agents, capable of dealing with real-world situations.

The Architecture proposed by Velásquez is extremely modular and establishes a very high-level relation between five groups of systems:

1. Perceptual Systems
2. Motor Systems
3. Behavior Systems
4. Emotional Systems
5. Drive Systems (Motivational System)

The actual implementation of these systems is based on a network of non-linear processing units called Basic Computational Units (BCU), which are composed by three elements:

1. a set of Inputs;
2. an Appraisal Mechanism;
3. a set of Outputs.

Higher level structures, such as the five groups of systems mentioned before, are built from the aggregation of these Basic Computational Units in complex networks. Interaction between high-level systems is also established by connecting the output of one or more BCU's from one subsystem to the input of BCU's in another subsystem. This way, the entire Architecture is a large network of BCU's.

The fundamental component of the Appraisal Mechanism inside BCU's is what Velásquez calls Releasers. Releasers are described as "computational units that filter sensory data and identify special conditions which will provide excitatory (positive) or inhibitory (negative) input to the system they are associated with" [Vel98a]. Releasers can bee regarded as functions capable of evaluating (sensory) input and generating an output signal that depends on the relevance of the input. There are two types of Releasers, the Natural Releasers, which are hardwired from development, and Learned Releasers that can be learned by associating certain stimuli with Natural Releasers.

3.3.2. Motivations, Drive Systems, and Emotional System

The Drive Systems that Velásquez proposes are conceptually similar (but not equivalent) to motivational systems: Drives are mechanisms that impel the Agent to Action. Drive Systems are composed by several BCU's, whose Releasers keep track of the value of several motivational variables in order to maintain them within a specific range. Whenever the value of one of these motivational variables falls outside the desired range, the appropriate Drive Releasers generate an error signal that will be used as input to other systems inside the Agent (e.g. Emotional and Behavior).

Velásquez draws a clear distinction between motivations, Drive Systems and Emotion Systems. Within the proposed Architecture, Emotion Systems are the main motivational forces that lead the Agent to Action. In fact, even Drive Systems use this function to promote specific behavior. As an example, an Agent is motivated to get food by a combination of Hunger (Drive) and Distress (Emotion) caused by Hunger.

3.3.3. The Emotional System

Velásquez also insists on combining both cognitive and non-cognitive factors of Emotion. He divides Emotional Releasers (i.e. Releasers of BCU’s inside the Emotional System) into four categories:

1. Neural. Neural Releasers are triggered by neuro-physiological conditions or compounds such as neurotransmitters, hormones, environment conditions, etc.
2. Sensorimotor. These Releasers are influenced by sensorimotor conditions, such as facial expression or body posture, that have the capability of triggering emotional mechanisms and eliciting Emotions.
3. Motivational. Includes all Releasers triggered by Motivational forces, which include both the Drive System and the Emotional System itself.
4. Cognitive. Cognitive Releasers account for Emotions triggered by cognitive processes, such as appraisal, comparisons, attribution, beliefs and memories.

All Neural, Sensorimotor and Motivational Releasers implemented in the Architecture are Natural Releasers (i.e. pre-wired). Cognitive Releasers were implemented as Learned Releasers. It is interesting to note that the Cognitive Releasers Velásquez had implemented in previous versions of the Architecture were Natural Releasers, based on Cognitive Appraisal theories. However, Velásquez reimplemented Cognitive Releasers using Learned Releasers because he considered that Cognitive Appraisal theories had limited capability to explain the underlying brain processes that support Emotion.

The Activation of BCU’s in Emotional Systems follows a different function from that of BCU’s in Drive Systems. It includes an extra term that accounts for the excitatory or inhibitory influence of other Emotional Systems, and also another term that introduces temporal decay behavior. Velásquez claims that these Activation functions enable the Architecture to support several different types of Emotional Phenomena, including:

- **Primary Emotions**. Primary Emotions are related with the activation of certain Emotional Systems, such as Disgust or Fear, by Natural Releasers. Primary Emotions are fundamental in providing the Agent with adaptation capabilities to immediate environment conditions. Velásquez brings the example of Fear Emotional System, which may be activated by a Natural Releaser that detects a dangerous situation, and generates the appropriate defensive context to cope with it.
- **Secondary Emotions**. As we have seen, Velásquez has devised special releasers called Learned Releasers that are capable of learning relations between certain stimuli and the activation of Natural Releasers. The activation of Learned Releasers can thus be considered as the mechanism that supports Secondary Emotions (Damásio). Secondary Emotions tend to emerge after Agents develop a certain environment experience and start developing associations between events, objects and Primary Emotions, which are always present because they are hard-wired. The new Learned Releaser will be able to influence the action selection in future situations. Whenever the Agent reencounters the emotionally tagged stimulus (the person), the Learned Releaser will trigger the associated Emotional response (Fear).
- **Emotion Blends and Mixed Emotions**. Although admitting that Emotion Blends and Mixed Emotions are not consensual matters within Emotion research, Velásquez claims that his Architecture may also support them [Vel98a]. According to Velásquez, in spite of the absence of an explicit model, Emotion Blends might emerge naturally by the simultaneous activation of two or more Emotional Systems. The co-activated Emotional Systems would subsequently be able to bias one or more non-conflicting perceptual or behavior systems.
- **Moods.** Velásquez follows the view that Moods differ from Emotions mainly in what concerns the level of arousal. In this perspective, Moods would be explained by a state of low activation of particular Emotional Systems. Emotional Systems in such a low activation state would have the potential to become highly activated even in response to smaller stimuli. Velásquez describes this behavior as being consistent with the established theories about Moods.
- **Temperament**. Architectural support for Temperament comes from the possibility of varying the parameters associated with Emotional Systems: thresholds, gains and decay rates. Velásquez gives the example of a “grumpy” Agent that would be configured by lowering the activation threshold and decay rate for Anger and increasing those for Joy. It is not clear, however, if such a variation is performed exclusively during the initial setup of the Agent, or if it may also be done automatically by the Agent during its interaction with the environment. This would represent another opportunity for adaptation.

3.3.4. The Implementation

Something quite interesting and original about Velásquez work is the fact that the proposed Architecture was implemented in software Agents and also on a hardware Robot, Yuppi [Vel98a]. Yuppi Sensory System is composed of several different sensors, including:

- Two CCD Cameras for Stereo Vision;
- Two Microphones for Stereo Audio;
- IR sensors for obstacle detection;
- Air pressure sensor for sensing touch (Pleasant or Painful sensation);
- Pyro sensor aligned to detect temperature changes (caused for example by the presence of people);
- A simple proprio-perception system;

Yuppi's Drive System is constituted by four different drives (three of which related to physical goals) that control internal variables:

- RechargingRegulation (Battery control);
- TemperatureRegulation (Temperature control);
- Fatigue (Energy Control);
- Curiosity (Interest control);

Yuppi's Behavior System includes 19 behaviors, most of them concerning the satisfaction of its needs. Some of Yuppi's behaviors are: *Search-For-Bone*, *Approach-Bone*, Cower, *Recharge-Battery*, *Wander*, *Startle*, *Avoid-Obstacle*, *Approach-Person* and *Express-Emotion*.

Emotional Systems include *Distress*, *Anger*, *Happiness*, *Disgust*, *Fear* and *Joy*. Yuppi's Emotional Systems support behavior-selection through simple Natural Releaser. They also enable the development of new emotional associations within each System through Learned Releasers.

Let us first address the role of Natural Releasers within Emotional Systems that are the basis of a set of Primary Emotions associated with Drive Systems, Sensory Systems and with environment interaction in general. For instance, unsatisfied Drives will activate directly (i.e. through Natural Releasers) both Distress and Anger Emotional Systems. Conversely, satiation of drives will activate Happiness, whereas Distress is also activated if Drives are over-satiated.

Sensory Information may also activate Emotional Systems. For example, every pink-reddish object spotted in the environment will activate Happiness, but yellow objects will activate Disgust. Darkness and some specific Blue objects will trigger Fear. Loud noises will activate Surprise. Interacting with people will also activate certain Emotional Systems. When people pet Yuppi, they promote a Pleasure sensation (Air Pressure sensor) that will activate Joy. On the other hand, disciplining action will cause pain and activate Fear.

Emotional Systems together with Drive Systems allow a simple selection of Behaviors. As an example, Velásquez describes a situation where the Activation level of Curiosity (Drive) reaches a high value, and Yuppi is motivated to wander around (Wander Behavior) looking for a pink bone. When Yuppi finds the pink bone the Happiness Emotional System will be activated which will promote the activation of behaviors such as Wag-Tail and Approach-Bone. However, if Yuppi fails to find the bone after a certain period of time, Distress (Emotion) will be activated which will then trigger the Droop-Tail behavior.

The previous examples were focused on the activation of Emotional Mechanisms via Natural Releasers, which give support to Yuppi's Primary Emotions. However, Yuppi can also develop Secondary Emotions through associations established by Learned Releasers. Another example given in [Vel98a] describes a situation where the Fear Emotional System acquires a new Releaser for loud sounds. In that situation, Yuppi is being disciplined and as a result is subjected to some Pain. Pain will promote the activation of Fear that in turn activates the Cower behavior. Initially, the loud sound does not activate the Fear Emotional System by itself, and therefore the Cower behavior will not be triggered. However, if both Pain and stimulus loud sounds start occurring simultaneously the Fear Emotional System develops a new (Learned) Releaser associated with loud sounds. After several simultaneous occurrences of both stimuli, the new Releaser will be able to activate the Fear Emotional System whenever loud sounds are sensed, promoting the subsequent activation of the Cower behavior. According to these results, Velásquez claims that the proposed Architecture is capable of supporting Emotional Conditioning, providing yet another mechanism for action-selection.

3.3.5. Overview and Comments

Velásquez proposed a simple and modular model for Emotional phenomena, following a strong neuro-biological inspiration. The model makes a clear distinction between Primary and Secondary Emotion. The author addresses mainly the problem of behavior selection within a set of possible behaviors. However, it is not clear if behaviors change their operating mode depending on their activation level of the Emotional system or of other behaviors. There seems to be an all-or-nothing policy regarding the behavior selection. Another question that remains unanswered is how to extend the influence of Emotional Systems to additional systems, such as perception, or to modulate cognitive tasks, such as planning or goal generation. We believe that a richer scenario would provide a better test-bed for the Architecture but, nevertheless, Velásquez ideas have proven to be very useful for our work.

3.4. Cañamero

Cañamero has developed extensive work [Cañ97] [Cañ98] [Cañ00] in this field. One of Cañamero's work main focuses is the problem of Action Selection in Autonomous Agents with multiple goals (concerns) [Cañ00].

Canãmero has developed an Architecture where Emotions play a key role. For the author, Emotions help reorganize goal priority in special circumstances and work closely with *Motivation* by amplifying their relative importance. Cañamero focuses her work on Primary/Basic Emotions which she defines as being those more closely related with survival and elementary goals. Primary Emotions usually have very strong physiological and expressive manifestations. Different (Primary) Emotions have resulted from a long Evolutionary path and help Individuals achieve Goals, closely related with self-survival, by changing the balance established between the Individual and its (Internal/External) environment. The author gives the following examples:

- Fear is related to protection from environment influences
- Anger is responsible for blocking environment influences
- Anxiety is concerned with diminishing the risk of dealing with unknown environments

Cañamero supports her study in the following hypothesis [Cañ98]: if Emotions play an important role in Biological Agents, increasing their Adaptation and Autonomy in complex, dynamic, partially controlled and uncertain environments, then, Artificial Agents operating in environments with similar characteristics will probably need similar mechanisms in order to be successful. Cañamero observes that Emotional Mechanisms provide an interesting Engineering solution, because they are able to interact with several behavioral and cognitive subsystems simultaneously. Furthermore, they are strongly connected with Goals rather than with particular behaviors, which contributes to an increased flexibility in dealing with the environment.

3.4.1. Emotion, Motivation and Behavior

Cañamero points out the following functional properties of Emotional mechanisms [Cañ00]:

- *Action Guidance and Motivation*. Contributes to categorizing events regarding goals. Change goal or motivation to deal with an urgent situation. Amplify the Effects of Motivation.
- *Bodily Adaptation*. Prepare the Individual to deal with Danger, unexpected events and opportunities.

- *Signalling Relevance of Events to Others*. Expression of Emotional State may be used as reference for social behaviors.

Cañamero defines a fundamental relation between Emotion, Motivation and Behavior in the context of Action Selection. Following the view of Fridja [Fri86], Ortony et al. [Ort88] and Rolls [Rol99] that relate Emotions with Goals, the author argues that the link between Emotion and Behavior is established through Motivation, instead of a simpler direct relation. This indirect relation is advantageous because it contributes to a more flexible and broad behavioral set.

Cañamero insists in distinguishing Emotions from Motivation or Drives, disagreeing with other authors [Pfe93] that propose a strong similarity between Basic Emotions and Motivation or Drives. Alternatively, Canãmero considers Emotions mechanisms that “modify” or “amplify” Motivation, therefore biasing the selection of behavior to be activated. Emotional Mechanisms continuously monitor internal and external environments to detect relevant events that require an appropriate response. This response is indirectly requested by a change in the Motivational System. Emotions are thus considered a *second order* control mechanism, capable of acting upon the motivational control mechanism and changing or resetting the Goal/Motivational priorities.

3.4.2. The Architecture

In [Cañ97], Canãmero presents an Emotional-based Architecture implemented in a simulated environment named *Gridland*. This Architecture has several interesting properties concerning learning and problem solving. However, we will try to focus more on the issues that are most relevant to the discussion of Emotional Mechanisms, namely on the relation between Emotion and Action Selection.

The *Gridland* environment is a two-dimensional grid populated by three types of elements:

1. *Creatures*, single-cell sized living beings. There are two different kinds of Creatures: Abbots and Enemies. Abbots are the Emotionally-enabled Agents. Enemies wander around Gridland, avoiding obstacles and eating food and drinking water whenever they find them. Enemies also try to eat Abbots, as well as each other. Enemies’ bites are a source of Pain.
2. *Resources*, namely Food and Water that exist as single-cell sized elements distributed in the grid. Whenever resources are consumed, they disappear and are regenerated in random positions.
3. *Obstacles* of varying size and shape that can be moved by Abbots.

Abbots have three types of sensors:

- *Somatic*, which provide information about their own body;

- *Tactile*, which allows the Agents to detect several properties of objects located in eight neighbor cells.
- *Visual*, which are capable of providing information about the brightness and distance of surrounding objects.

Abbots have a set of three Effectors: hand, foot and mouth. These Effectors support several possible behaviors, such as *drinking*, *eating*, *walking*, *avoiding obstacles*, *playing*, *attacking*, *withdrawing*, *look-for*, *look-around*, etc. Whenever an Abbot attempts to execute behaviors contrary to the physical constraints of the environment, it will feel pain. Abbots will also feel pain when Enemies bite them.

Canãmero argues [Cañ00] that the *Gridland* is a suitable environment for the study of Action Selection and Emotional Mechanisms because it contains the following set of characteristics:

- *Highly Dynamic,* since objects change their locations frequently;
- A reasonable level of *Uncertainty,* resulting from noisy perception and the intrinsic dynamic of the Environment;
- Existence of some *Social Interaction* as Abbots and Enemies compete for the same resources and engage in "flee or fight" situations. However, there is no cooperation or social attachment between Abbots.
- *Multiple Threats in the Environment*, external (Enemies and Angry Abbots) and internal (physiological level reaching dangerous threshold levels).
- *Randomly Distributed Resources* (water and food) that change their position whenever they are consumed and that might, therefore, be unavailable in the Agent's surroundings.

The author proposes an *explicit* model for Emotional Mechanisms [Cañ98], which, as we have seen before, she considers to be the most appropriate strategy for designing Emotional Mechanisms. Therefore, Emotional Mechanisms have the following "explicitly engineered" properties:

- a triggering event;
- an intensity level;
- an activation threshold;
- a list of Synthetic Hormones to be released when activated;
- a list of Physiological manifestations;

Abbots have six Emotional primitives that address needs, mostly related with survival functions. Abbots Emotional set includes:

1. *Anger*: Mechanism that is triggered when the accomplishment of a goal is menaced. Anger is a mechanism capable of blocking influences from the environment.
2. *Boredom*: Mechanism intended to stop a repetitive behavior that is proving to be inefficient. The triggering event is a prolonged inefficient behavior.
3. *Fear*: An Emotional state triggered by the presence of Enemies. It functions as a Defensive Mechanism.
4. *Happiness*: Mechanism intended to obtain re-equilibration after the accomplishment of a Goal.
5. *Interest*: Triggered by the presence of a novel object in order to stimulate the interaction with it.
6. *Sadness*: Mechanism that is triggered by the inability to attain a specific goal and results in a global reduction in the activity of the Abbot. "Sad" Abbots enter a suspended state waiting for the occurrence of changes in the environment or in their internal state.

Most of the Emotional primitives chosen by Canãmero can be considered as corresponding to Primary or Basic Emotions, namely "Fear", "Anger" and "Happiness". However, the other Emotional primitives - "Boredom", "Interest" and "Sadness" -, in the way they are defined, are, in our opinion, less prone to be regarded as Primary or Basic Emotions, in spite of their undeniable functionality.

Behavior selection in Abbots is achieved by the combination of both Motivational System and Emotional Mechanisms. Abbots have an internal state comprising a set of Homeostatic variables that build the core of the Motivational System. The Homeostatic variables correspond to the Abbot's main Motivations, namely aggression, cold, curiosity, fatigue, hunger, self-protection, thirst, and warmth. Whenever Homeostatic Variables fall outside a specific range, the Motivational Mechanisms will try to activate the behavior that best satisfies the most urgent need.

Emotional Mechanisms work in parallel with the Motivational System trying to increase the flexibility of behavior selection and execution. In fact, Emotional Mechanisms monitor the environment (internal and external) for significant changes or patterns that will then trigger the activation of one or more Emotions. Active Emotional Mechanisms release the corresponding Synthetic Hormones that are capable of changing the value of Homeostatic Variables, as well as Abbots' Perception System.

A change of Homeostatic Variables represents a change in the Agent's global Motivational System. This results either in the promotion of a specific behavior or in the interruption of current active behavior in order to react appropriately to an environment opportunity or threat. However, it is also important to note that the Hormone Level also influences the execution of the selected behavior. Behaviors have configurable parameters, such as duration or intensity, which also depend on the Hormone Level, and can effectively be customized to adapt to environment conditions.

Emotional Systems have thus a twofold influence over behaviors. Firstly, Emotional Mechanisms exert indirect influence on behavior selection by biasing Abbots' Motivational state. Secondly, the Hormonal context Level they generate is capable of directly shaping the way a specific behavior is executed after being selected.

Abbots' perception is also influenced by their Emotional Mechanisms, although Cañamero does not explore this issue in great depth. Abbots' Memory System is based on an ART-1 neural network [Cañ97] whose capability to store and recognize objects depends on the "Vigilance Threshold" parameter that varies with the current Emotional state. Very intense Emotional States, corresponding to an "alertness" state, will promote a finer categorization of objects to be stored or recognized, whereas intermediate intensity Emotional States will lead to a coarser granularity of the categorization. Additionally, Emotional States also alter Abbots' proprio-perception by producing hormonal modifications. For example, under "Happy" Emotional State, the Endorphine hormone is released, reducing the perception of pain.

3.4.3. Evaluation

Another important issue that Cañamero brings to discussion is the need for established means to evaluate the influence of Emotional Mechanisms in the Agent's performance [Cañ98]. As any other evaluation activity, evaluating Emotional Mechanisms may be extremely complex but, as pointed out by the author, only by defining common metrics, evaluation criteria and test-beds will it be possible to track objectively the development of Emotional Agents and compare different Emotional Agent Architectures. In our opinion, this is certainly a very significant subject, as it can be easily confirmed by the amount of different Architectures proposed. In fact, research in MAS in general and Emotional Agents in particular suffers from a great deal of subjectivity and from a lack of consistent concept definitions.

3.4.4. Overview and Comments

The work provided by Cañamero presents a very functional perspective on Emotion, showing a clear connection between Emotion, Goals and Behavior. An interesting point about Cañamero's work is that the influence of Emotion on Behavior resides both on the *selection* of the active Behavior and also on its *modulation*, according to some specific parameters. In fact, this possibility is also extended to perception by introducing the notion of "vigilance threshold" that alters the detail at which sensory data is processed. We suppose that the potential of Cañamero's architecture and ideas could have been more evident if a more complex simulation environment had been used. Nevertheless, we think the ideas described have provided good orientations for our own work.

3.5. Gadanho

Another interesting approach to the use of Emotional-like mechanisms has been pursued by [Gad98] and [Gad00] which combines Emotions with Adaptive Control and Reinforcement Learning.

In [Gad98], the authors address the problem of designing an Adaptive Controller for a simulated Robot, operating in a continuous real-time environment. The Perception of the simulated Robot employed in their experiments comprises a set of eight sensors, which allow it to detect obstacles in its surroundings, and light sensing capabilities, with which the robot can detect light points in the environment. Using two side motors for locomotion, the robot has to make its way in a labyrinth-like environment, delimited by walls and obstacles, searching for valuable energy that it needs in order to survive. The Robot is constantly consuming energy from its limited-life battery that can only be refilled at specific points available throughout the environment and identified by a detectable light. However, these energy refilling points have themselves limited capability and require a considerable recharge time before they can be used again by the Robot. Therefore, the Robot must constantly search for the best alternative energy points available. The simulation system introduces a certain percentage of noise in the values read by the sensors and in the effective power delivered to the motors, creating supplementary problems in both Perception and Control. The Robot has three pre-programmed behaviors available that build its (compound) Action/Behavior set:

1. Avoid Obstacles
2. Seek Light
3. Follow Walls

Gadanho and Hallam's goal is to develop a controller that is able to optimally choose the appropriate behavior from the three possible behaviors available, according to the current state of the environment. The controller is built using a Reinforcement Learning approach, in particular Q-Learning.

However, an environment such as the one described before (real-time, continuous, dynamic) creates a serious difficulty for the convergence of Learning algorithms as well as for the global efficiency of the Controller itself. In fact, a continuous environment imposes the intrinsic difficulty of defining state transitions, since a new (multi-dimensional) state can be found at virtually any step. For this reason, the authors devised a specific Agent Architecture that explicitly uses Emotional concepts to tackle the problem just described. The Emotional mechanism developed involves the following concepts and organization:

- *Sensations*: values obtained by using data collected from the Robot's sensors and also from other events calculated using the Robot's internal variables (e.g.: level of actuators).
- *Feelings*: values obtained by combining the value of Sensations with the value of Hormones.
- *Emotions*: The level of each Emotion results from combining the values of several Feelings. Although there are several Emotions at stake, the robot behavior will be defined by just one of them that is considered to be the *Dominant Emotion*. The *Dominant Emotion* is selected according to pre-defined threshold levels at every step of the simulation.
- *Hormone System*: a System that combines the values of several Emotion levels and produces Hormones. These Hormones will be used to influence the level of the Feelings in combination with Sensations. The Hormone System acts as feed-back loop and it helps stabilize the value of Feeling and Emotions, in spite of possible quick changes in Sensations' values.

Using this architecture, the problem of state transition was solved by triggering transitions not directly from sensory information, which changes too quickly and is prone to instability, but from changes in Emotional levels instead. When the level of one Emotion suffers a change higher than a predefined threshold value, a new state transition is triggered.

The authors have successfully employed this strategy in training a controller for the Robot, using Q-Learning. The Q-Learning algorithm was capable of converging and produced an efficient controller. The authors have also compared the performance of the controller developed using emotion triggered state transition with controllers developed using other strategies. Specifically, they have developed another controller using Q-Learning, but used a fixed interval of time to trigger state transitions. According to the authors, these two controllers showed similar performances in controlling the Robot, but the one developed using Emotion-triggered transitions was able to converge in significantly less steps (one sixth of the steps). This reduced computational effort is extremely important in real-time control situation such as the one in the experiment, because it releases processor power to other concurrent processes, if needed.

In [Gad98], the authors also suggest a strategy that could be employed to reduce the number of steps needed to train a controller that uses time triggered state transition. Since Emotional Mechanisms should be able to reflect the relevance of the environment, (we will address the problem of choosing Emotion and their relations later) the controller should eventually learn more from situations where Emotions have substantial activation levels. Based on this Emotion Level/Environment Relevance correspondence, which is fundamental in Emotional Architectures, it seems that reducing the time between state transitions whenever the level of Emotion is high, i.e. increasing the frequency of state transitions, may be advantageous for the learning process. Conversely, reduced levels of Emotion should signal less relevant periods in the environment and should not require such frequent state transitions. This decreased frequency would allow the controller to converge in less steps and release processing resources.

3.5.1. Comments and Overview for Gadanho

This work shows a very interesting application of Emotional mechanisms to Autonomous Agents, specifically in connection with learning algorithms in complex environments. Two possible connections between Emotional mechanisms and learning algorithms are considered. In the first case, Emotional mechanisms are used to define the State of the environment to be learned, instead of using only sensory information. The author reports that this strategy actually results in faster learning times by reducing the influence of noise in data and the dimension of problem. This is a clear example of how to use the *Emotion as Information* paradigm (see Chapter 2). In the second case, the author proposes using Emotional mechanisms in order to adjust specific parameters of the learning algorithm (namely cycle of state transition), thereby achieving shorter convergence times. This is based on the assumption that relevant information should be signaled by intense Emotions and may be seen as an example of *Emotion as a mechanism for Operating Mode configuration* (also Chapter 2).

3.6. Summary and Conclusions

In this Chapter, we made a brief review of some of the works that influenced most the development of our own ideas. The authors share the notion that Emotions are phenomena tightly connected to Agents that operate in complex worlds, and that help them in dealing with such complexity. Most of the work presented was concerned with the problem of Action-Selection and adaptation to the environment, especially the work by Velásquez and Cañamero. Emotional Mechanisms were involved in changing or complementing the Agent’s basic motivations that lead to action. Emotional mechanisms were also used as a way to control resources and schedule internal processing (interrupt, swap). This was shown to be particularly useful for Agents whose processing resources are limited and that have to deal with multiple and simultaneous Goals. We have also seen how Emotion could be used in improving Agent Learning capabilities by providing an alternative source of information for the learning process. Emotional Information is more stable and more compact than the information available exclusively through sensors, speeding up the learning process. Additionally, the process of learning itself could be modulated according to the activity of Emotional Mechanisms in order to improve convergence.

All these Architectures seem to have a complementary view about the functional role of Emotion. However, in most cases the authors focus only on a particular possible use of Emotion, and do not include other interesting perspectives that have been identified throughout the literature. We believe that there is still a lot of work to do in that direction, and that this is a challenging endeavor for AI.

CHAPTER 4. MODELING EMOTION MECHANISMS

4.1. Introduction

In this chapter, we will try to formalize and model the concepts presented in Chapter 2, related both with Emotion and Emotion-Cognition interactions, with the purpose of enabling their integration within Agent Architectures.

We will start by revisiting Emotional Elicitation as the first step in the process of modeling Emotional Mechanisms. We will try to integrate Emotional Mechanisms within other global concepts of Agent Architectures, namely Goals, Perception, *Internal State* and Proprio-perception. We will refocus on the concept of Emotional Evaluation Function, briefly presented in Chapter 2, and show its relation to Emotional Elicitation.

Next, we will once again address the concept of Emotional State and introduce the notions of Emotional Accumulator and *Basic Emotional Structures*. These two concepts together with the Emotional Evaluation functions are the basic components of our Architecture.

We will finish this Chapter presenting four models of Emotional Interactions, in particular, those regarding the concepts of:

- Emotion as Information;
- Emotion as a Process Control mechanism;
- Emotion as a Resource Allocation mechanism;
- Emotion as a mechanism for defining Agent's Processing Strategies.

Through these models, we formalize all important concepts relevant for our approach.

4.2. Elicitation revisited

In Chapter 2 we described the process of Emotional Elicitation by which Emotional phenomena are activated or triggered. The process of Emotional Elicitation was basically concerned with the evaluation of the chances of a certain Goal being achieved (or not), considering both the state of the environment (external state) and the Agent's internal state. We will now take a closer look at this subject.

4.2.1. Multiple and Simultaneous Goals

To conveniently model the process of Emotional elicitation, we will first need to consider some issues about Goals in complex environments. Let us first assume that an Agent operating in a complex environment will have *multiple* goals. This is a reasonable assumption, even if the Agent has only one basic fundamental goal: for that goal to be achieved, at some point in time, it will be necessary to unfold it into multiple sub-goals. Additionally, since the environment is only partially controlled by the Agent, situations representing new threats or imposing restrictions on current goals will certainly occur, motivating the Agent to create new Goals. Of course, such situations may also represent interesting opportunities for the achievement of certain Goals and, therefore, the Agent will generate new Goals to address those opportunities. In any case, it is perfectly reasonable to assume that the Agent will have multiple goals.

Furthermore, we may also safely assume that in complex environments Agents will have *simultaneous* Goals. The environment evolves even if the Agent does not perform any action. More than one threat or opportunity may occur close in time. Therefore, even if the Agent is developing efforts to achieve a specific important Goal (which he must not abandon), situations that generate a new urgent Goal will eventually occur, forcing the Agent to deal with more than one goal at the same time.

However, not every Goal needs to be immediately considered by the Agent. Goals have different priorities and distinct levels of importance for the Agent. Some goals have a high priority level while others may be postponed for some time or simply discarded. In addition, goals may have dependencies (sub-goals) to be observed which alters the priority of their execution. We will now try to formalize these concepts.

Let g_i denote a specific Agent Goal at instant *t*. Each goal g may be assigned a *priority,* p_i, and a *Set of Dependen*cies, D_i, variable over time and composed of individual dependencies d_{ik}:

$$g_i(t) = \{p_i(t), D_i(t)\}$$

$$D_i(t) = \{d_{i1}(t), d_{i2}(t)\ldots\ d_{in}(t)\}$$

More urgent goals will have higher levels of priority. Dependencies d_{ik} establish relations between one goal g_i and other goals. This implies that the achievement of Goal g_i may only be reached when no more dependencies exist, i.e. the dependent goals have been achieved:

$$achievement(g_i) \Rightarrow D_i(t) = \varnothing$$

Let [G(t)] denote the *Set of Goals* that the Agent is pursuing at instant t. Then:

$[G(t)] = \{g_1(t), g_2(t)\ldots g_n(t)\}$

where $g_i(t)$ is a specific Goal that the Agent possesses at instant t. The *Set of Goals* is never an empty set because the Agent will always have at least one Goal, the *Fundamental Goal,* g_F:

$\forall\ t, [G(t)] \neq \varnothing$

or, more specifically,

$\forall\ t, [G(t)] \supset \{g_F(t)\}$

The *Fundamental Goal* is a Goal that exists during the Agent's entire lifespan and its existence is intrinsically connected with the Agent's own existence. Any other Goal requires (either implicitly or explicitly) the Fundamental Goal to be ensured, implying that some (or all) of Agent's Capabilities and Resources need to be constantly allocated to the achievement of the *Fundamental Goal*. The Fundamental Goal does not change during Agent's existence. However, depending on the specific environment conditions the Agent may spawn different sub-Goals that contribute to the achievement of the *Fundamental Goal*.

The *Fundamental Goal* has priority p_F and may have a non empty Set of Dependencies D_F. No goal has higher priority than p_F:

$\forall\ g_i \in [G(t)] : p_i \leq p_F$

However, the priority of Goals belonging to the *Set of Dependencies* of the *Fundamental Goal* g_F is equal to p_F, since all dependent Goals must be accomplished previously to g_F. Let as also introduce the SG operator. The SG returns the number of *Simultaneous Goals* occurring in a given Goal Set (at instant t):

$n = SG([G(t)])$

For Agents operating in complex environments, SG([G(t)]) is almost always larger than one because the Agent will have at least the *Fundamental Goal* and the others resulting from the interaction with the environment. The Agent will be considering simultaneous goals by their priority.

However, since goals cannot be permanently postponed, Agent *Capabilities* and *Resources* will need to be shared between the processes leading to their achievement.

4.2.2. The Perception of Environment

In order to have any chances of being successful, an Agent will need to capture a reasonable set of significant environment features. Each feature may be obtained either by direct inspection of sensory data or it may result from a more complex analysis, perhaps using previously stored information. In any case, the Agent will need to create a model of the external environment using such features. Let [E] denote the *Environment Model* that the Agent is able to obtain using a given set of significant features (extracted from the environment or produced). Then [E] is composed by a set of Beliefs[5], e_i, about multiple elements existing in the environment.

$$[E(t)] = \{e_1(t), e_2(t)\dots e_n(t)\}$$

Since sensory data is intrinsically noisy and the Agent has limited resources to analyze it, [E(t)] is only a rough approximation of the actual environment state. However, up to this moment, it is all the Agent has to derive its decisions and actions.

One should note that not every Belief about the environment is relevant for all of the Agent's Goals. For each goal g_j belonging to the *Set of Goals* [G(t)], there is a subset of [E(t)], denoted by $[E(t)]_j$, so that all elements of $[E(t)]_j$ are relevant for the achievement of goal g_j:

$$\forall\, g_i \in [G(t)]\ \exists\ [E_i(t)] \subset [E(t)] : e_k \in [E_i(t)] \Rightarrow \mathrm{isRelevant}(e_k, g_i)$$

where the relation isRelevant(e, g) is true if *e*, a specific Belief about the Environment, has information that is necessary for the achievement of Goal *g*.

For each goal g_i, it should be possible to define a matrix, the Relevance Matrix Rl_i, that relates [E(t)] with the corresponding $[E_i(t)]$:

$$[E_i(t)] = Rl_i \bullet [E(t)]$$

As described in Chapter 2, Emotional Elicitation operates over a subset of [E(t)] that is relevant for a given goal, in order to detect situations that pose a threat (or opportunities) for the achievement of each goal. In a certain way, Emotional Elicitation processes should be able to perform relevance analysis as if they had intrinsic knowledge of the Relevance Matrix.

---

[5] A Belief can be defined as a piece of information gathered through perception filters, produced by internal information processing mechanisms or obtained from other Agents not yet proved to be true. In the context of some Intentional Logics, Beliefs that are proved to be true become Knowledge.

Finally, since it should be possible to obtain a Relevance Matrix for each Goal, we may define the Set of Relevance Matrixes [Rl(t)] as:

$\forall g_i \in [G(t)] \; \exists Rl_i : Rl_i \subset [Rl(t)]$

Almost certainly, Agents will not know accurately each Rl_i because the environment is complex and they have limited sensory and analysis resources. Consequently, they will only be able to obtain an estimate of the Set of Relevance Matrixes, [Rl'(t)]. We will pick up this issue again in the following sections.

4.2.3. Internal State of the Agent

It is easy to understand the direct importance of the perception of environment state, [E(t)] in the process of Emotional Elicitation. However, we also need to capture the importance of the Agent's *Internal State*.

Let us consider that an Agent possesses a *Set of Capabilities*, [C(t)], that enable it to operate in the environment. These *Capabilities* include:

- information processing mechanisms (e.g.: planning, inference, communication, algorithms);
- action mechanisms capable of changing the environment up to some extent (e.g.: change its position or the position of other objects);

We may assume that over time the *Set of Capabilities* may change as the Agent may acquire or lose certain *Capabilities*. However, the actual instantiation of these *Capabilities* is always bounded by the *Set of Resources*, [R(t)], that the Agent has available either intrinsically or at a given instant. *Resources* include:

- the computational structure that supports information processing mechanisms (memory, computational power)
- the physical (virtual) infrastructure that enables Agents to perform actions in the environment (e.g.: muscles, robotized arms, tools or machines that help in the execution of tasks)
- other Agents that may help in the achievement of a given goal;
- any resource that may be used to obtain, maintain or extend any of the previously mentioned resources.

Therefore, we may define the Set of the *Effective Capabilities* of the Agent, [EC(t)], as the *Set of Capabilities* that the Agent may effectively use considering the *Resources* available at that moment and their allocation:

[EC (t)] = A([G(t)],[C(t)], [R(t)])

where A is a *Global Allocation Function* that assigns *Resources* to *Capabilities* according to the *Set of Goals*. The *Global Allocation Function* is able to globally coordinate the allocation of Resources among Agents Goals. Although the *Global Allocation Function* influences all *Capabilities*, it is not intended to provide a fine-grain control over each Capability. However, it may generate many different allocation contexts so that Agent *Capabilities* may be instantiated in multiple ways. Particular allocations of *Resources* to *Capabilities* will be related with different *Process Operating Modes* and a distinct global *Processing Strategy.*

Let us then define the Agent's *Internal State* [I(t)] as the combination of all these sets in order to explicitly include the set of *Effective Capabilities* at a given instant:

[I(t)] = {[C(t)], [R(t)], [EC(t)]}

4.2.4. Proprio-Perception

It is not guaranteed that the Agent knows exactly its own *Set of Capabilities* [C(t)] or its Set of *Resources* [R(t)]. Instead, the Agent has an estimate of all these sets over time, which we will denote respectively by [C'(t)] and [R'(t)]. In fact, for sufficiently complex Agents, the Agent's entire *Internal State* might only be known approximately, even by itself. The Agent may estimate its own Set of *Effective Capabilities* taking:

[EC (t)] = A'(t, [G(t)], [C'(t)], [R'(t)])

where A' is an estimated *Global Allocation Function* that corresponds to the Agent's estimate of how it might allocate its own perceived *Resources* to its perceived *Capabilities.*

Let us now define Agent *Proprio-Perception* [P(t)] as the combination of the previous sets, the Modified Allocation Function, A', and also an estimate of the *Set of Relevance Matrixes* defined in a previous section:

[P(t)] = {[C'(t)], [R'(t)], [EC'(t)], A', [Rl'(t)]}

We may also *extend* our previous definition of *Internal State* in order to include Agent's *Proprio-Perception*:

[I(t)] = {[C(t)], [R(t)], [EC (t)], [P(t)]}

We have now defined and formalized all concepts needed for a systematic notion of Emotional Elicitation.

4.2.5. Emotional Evaluation Functions

In Chapter 2, we established a relation between Emotional Elicitation and the evaluation of chances of Goal achievement, taking into account the perception of the environment and the *Internal State* of the Agent (effective and perceived). The key element in this process is the Emotional Evaluation Function that is able to evaluate these parameters and promote a change in the *Emotional State* of the Agent[6]. We shall now formalize this concept.

Definition

Let [E(t)] be the *Environment Model,* as perceived by the Agent, and [I(t)] be the Agent's *Internal State,* which includes both the effective and perceived state. Then, for a g_i belonging to the *Set of Goals*, [G(t)], there is an *Emotional Evaluation Function*, EEF, that is able to produce a change, [ΔEm], in the *Emotional State* of the Agent, [Em]:

$$[\Delta Em] = EEF(g_i(t), [G(t)], [E(t)], [I(t)])$$

Note that we do not impose any restriction regarding the nature of the Emotional Evaluation Function itself: it may be a simple algebraic function that maps a combination of input values to a scalar output value (similar a neurobiological / sensory circuit),or it may quite possibly be a very complex inference or pattern matching procedure (similar to a high-level cognitive analysis).

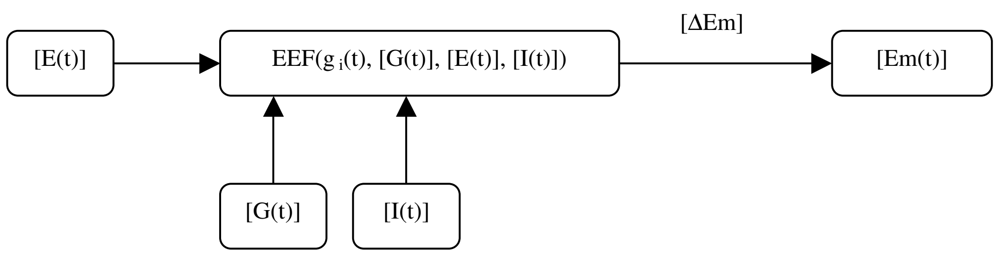

Figure 4.1 – Emotional Evaluation Functions produce changes in the Emotional State

Expanding each of the parameters of EEF we obtain:

---

[6] A deeper formalization of the concept of Emotional State will be developed later.

$[\Delta Em] = EEF(g_i, [G(t)], [E(t)], [I(t)]) =$

$EEF(\{p(t), D(t)\},$

$\{g_1(t), g_2(t)\ldots g_n(t)\},$

$\{e_1(t), e_2(t)\ldots e_n(t)\},$

$\{[C(t)], [R(t)], [EC\,(t)], [P(t)]\})$

And now expanding [P(t)] we obtain:

$[\Delta Em] = EEF(\quad \{p(t), D(t)\},$

$\{g_1(t), g_2(t)\ldots g_n(t)\},$

$\{e_1(t), e_2(t)\ldots e_n(t)\},$

$\{[C(t)], [R(t)], [EC\,(t)], \{[C'(t)], [R'(t)], [EC'(t)], A', [Rl'(t)]\}\})$

We may now verify that *Emotional Evaluation Functions* focus on the success chances of a given goal considering:

- the Goal's priority and dependencies
- other Goals at stake (priorities, number of simultaneous Goals)
- the perception of multiple elements of the state of the environment $\{e_1(t), e_2(t)\ldots e_n(t)\}$
- the Agent's *Set of Capabilities*, Set of *Resources* and Set of *Effective Capabilities*
- Agent's perception of the previous Sets
- Agent's perception of the Relevance of multiple elements of the environment

The change in the *Emotional State* produced by the EEF has three main purposes:

1. signaling the Agent the relevance of a particular environment situation for one of the Agent's goals (i.e. produce information)
2. promoting a certain allocation of *Resources* to *Capabilities* in order to generate the appropriate operating context (i.e. manage resources)
3. change the way certain information processing mechanisms are operating in order to adapt them to current circumstances (i.e. change operating modes)

The Emotional Evaluation Function (EEF) is, however, a very general mechanism leaving much to be said about the process of evaluation itself. Most of the modeling effort presented here is intended to provide a basic framework for developing Agents with emotional-like mechanisms.

4.3. Emotional State

4.3.1. Redefinition

We will now focus again on the concept of *Emotional State* and try to extend and formalize our previous definition given in Chapter 2.

Let us consider that we augment the Agent's *Internal State* with a set of continuous non-negative variables, [Em], that vary over time:

$[Em(t)] = \{em_1(t), em_2(t)\ldots\ em_n(t)\}$[7]

$\forall\ em_i \in [Em(t)], \forall\ t\ 0; em_i(t) \geq 0$

The variables em_i may be directly updated by Emotional Evaluation Functions, which are able to change their values with positive or negative increments:

$[\Delta Em] = EEF(g_i(t), [G(t)], [E(t)], [I(t)])$

$[\Delta Em] = \{\Delta em_1, \Delta em_2\ldots\ \Delta em_n\}$

The values of each individual component, $em_{i,}$ decay with time with a specific rate, which may be different for each component. Therefore, unless the value em_i is updated by an EEF, it will decrease monotonically with time.

The values of the variables em_i are globally accessible by other elements of the Agent's *Internal State*, although they may not be changed directly by those elements. Such set of values, [Em(t)], is called the *Emotional State* of the Agent. Again:

$[Em(t)] = \{em_1(t), em_2(t)\ldots\ em_n(t)\}$

4.3.2. Further Implications

After defining the concept of *Emotional State* we must now redefine some related concepts. Let us first redefine the Agent's *Internal State* in order to include the Emotional State also:

---

[7] We will avoid the temptation of referring to each of these individual variables as "Emotions", although this is obviously the corresponding anthropomorphic concept at stake.

[I(t)] = {[C(t)], [R(t)], [EC (t)], [P(t)], [Em(t)]}

Since the *Emotional State* is now included as a parameter in the *Internal State*, it becomes an input for EEF's, thereby creating a direct feedback loop. This opens the possibility for very complex interactions.

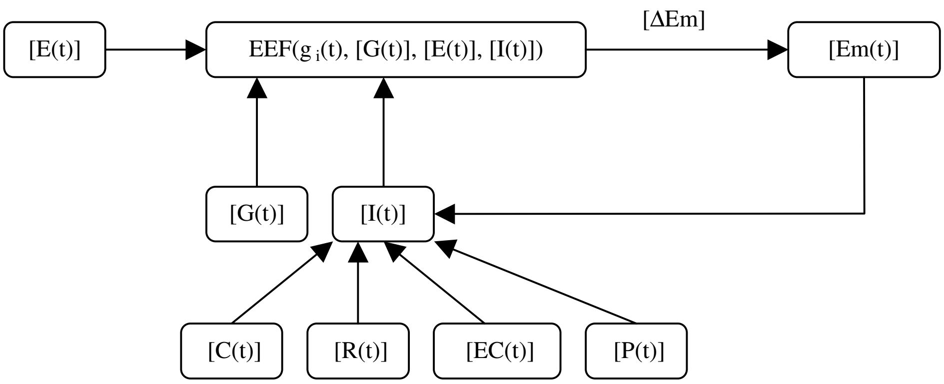

Figure 4.2 – A direct feedback loop between the Emotional State and Emotional Evaluations functions is established through the Internal State [I(t)] of the Agent, which includes the Emotional State.

Additionally, an estimate of the *Emotional State* should also be included in Agent's *Proprio-Perception*. Although some Emotions may not be consciously felt or differentiated, we may assume, nevertheless, that part of them are still perceived either directly, or indirectly (by their consequences and effects). Let us then assume that [Em'(t)] is the Agent's estimate of its own *Emotional State* [Em(t)]. The Agent's *Proprio-Perception* Pi may now be extended to include such estimate [Em'(t)]:

[P(t)] = {[C'(t)], [R'(t)], [EC'(t)], A', [Rl'(t)], [Em'(t)]}

Again, it should be noted that this modification in Agent's *Proprio-Perception* creates an additional feedback loop around EEF's that complicates possible interactions even further.

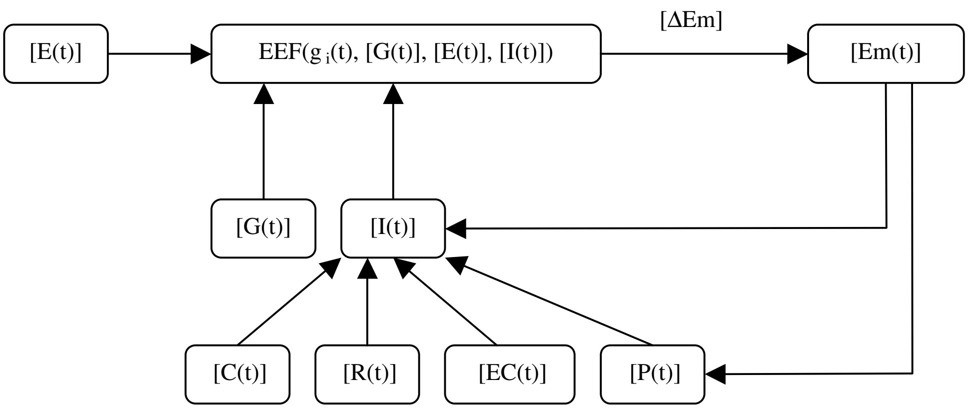

Figure 4.3 – Another feedback loop the Emotional State and Emotional Evaluations functions is established through Agent's Proprio-Perception of its own Emotional State.

4.3.3. Emotional Accumulators and Basic Emotional Structures

As we have mentioned before, each of the variables em_i that compose the *Emotional State* has a time dependent evolution that may be different from other variables. We shall now introduce another concept that will allow us to modulate such a behavior, the *Emotional Accumulator* [Oli02][Oli03].

Definition:

An *Emotional Accumulator* (EA_i) is a time (t) dependent process that incrementally stores a percentage ($P_{input,}$ - *Input Percentage*) of an input value, i, obtained from any of several Emotional Evaluation Functions to which it is connected (by default, only one EEF). The value stored by an *Emotional Accumulator* decays exponentially with a specific time constant (T_d - *Decay Time Constant*):

$$EA_i(t, i, P_{input}, T_d)$$

The practical application of *Emotional Accumulators* is to store the values of the variables that compose the Emotional State. In other words, the Agent's *Emotional State* is stored by a set of *Emotional Accumulators*, one for each of the Emotional State's variables:

$$em_i(t) <=> EA_i(t, i, P_{input}, T_d)$$

Then:

$$[Em(t)] <=> \{ EA_1(t, i, P_{input}, T_d), EA_2(t, i, P_{input}, T_d) \ldots EA_n(t, i, P_{input}, T_d) \}$$

Emotional Accumulators work in connection with Emotional Evaluation Functions creating the basic structure of our architecture: the Basic Emotional Structure. The following picture tries to depict this association.

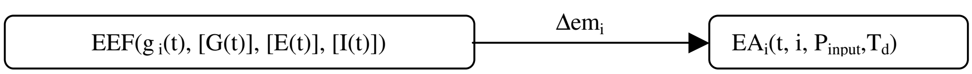

Figure 4.4 – The Basic Emotional Structure: Emotional Evaluation Function – *Emotional Accumulator* connection.

Basic Emotional Structures are capable of providing a time behavior compatible with what was specified in the previous sections. *Emotional Accumulators* will store positive increments coming from connected EEF's and impose a monotonic decay towards zero. The next picture illustrates this process:

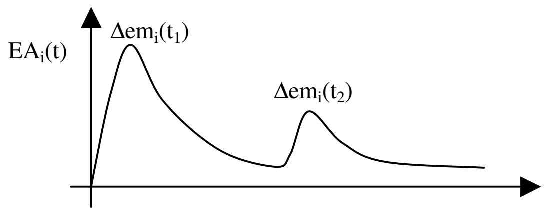

Figure 4.5 - Time behavior of an *Emotional Accumulator*. Increments represent updates from the coupled EEF. Accumulator value decreases at a rate specified by T_d.

By controlling the two essential parameters of the *Emotional Accumulator*, P_{input} (*Input Percentage*) and T_d (*Decay Time Constant*) it is possible to model different types of *Emotional Phenomena,* as discussed in Chapter 2: *Specific Emotions*, *Moods* and *Emotional Dispositions*. The P_{INPUT} (*Input Percentage*) parameter controls the influence of the EEF in the increment of the Accumulator while the T_D parameter (*Decay Time Constant*) will define how long the Emotional Structure stays active. Small values of P_{INPUT} will slow the elicitation times (as in Emotional Dispositions). Small values of T_d will make the value of the *Emotional Accumulator* decay quickly like what happens with Specific Emotions. The next table summarizes these possibilities:

Table 4.1. Ranges of P_{INPUT} and T_D parameters

	P_{INPUT}	T_D
Specific Emotions	High	Low
Moods	Medium	Medium
Emotional Dispositions	Low	High

We have now formalized and modeled the essential properties of *Emotional Phenomena* and presented simple mechanisms capable of implementing such models. However, we have just been focusing on individual *Emotional Phenomena* and we have not yet formalized nor modeled how such individual phenomena may functionally interact with the other processes of the Agent. We will address this subject in the next section.

4.4. Modeling Emotional Interactions

4.4.1. Modeling Agent Processes

It is reasonable to assume that an Agent that operates in a complex environment should be equipped with various capabilities. As mentioned before, these will compose its *Set of Capabilities*. Let us also assume that in order to implement each individual *Capability,* the Agent will run one or more *Processes,* whose execution will contribute to such *Capability*. A *Capability* may be implemented by a single process or it may be accomplished by running more than one Process, either in sequence or in parallel.

Let P_i be a *Process* that is required for the accomplishment of a given *Capability* C_i. *Process* P_i will receive a set of *Input Arguments*, $[IA_i]$, and will produce a set of *Output Arguments* $[OA_i]$ and, possibly, the *Capability*, C_i.

$$P_i([IA_i]) \text{ -> } \{[OA_i], C_i\}$$

Process P_i may also receive a set of *Control Arguments*, $[CA_i]$, that control its execution. Depending on the specific *Control Arguments*, *Process* P_i will generate (or contribute to) different instantiations of the *Capability* C_i.

$$P_i([IA_i], [CA_i]) \text{ -> } \{[OA_i], C_i\}$$

The next picture illustrates these mechanics:

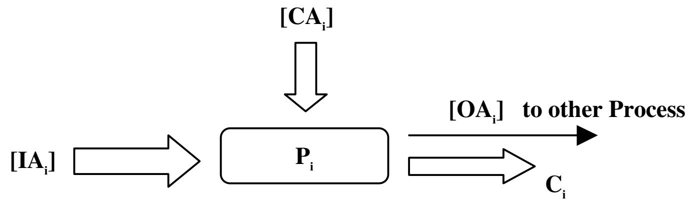

Figure 4.6. A Cognitive Process P_i receives a set of Input Arguments $[IA_i]$ and is able to produce a set of Output Arguments $[OA_i]$ (which may be fed as input arguments to other Processes) and, possibly, a Capability C_i. The actual instantiation of the Capability depends on the values of Control Arguments $[CA_i]$.

The set of *Control Arguments*, $[CA_i]$, will essentially modulate *how Capability* C_i is to be obtained through *Process* P_i but will not interfere fundamentally with it. The interaction with *Capability* C_i will be made indirectly. *Control Arguments* change processing parameters such as the complexity and the maximum running time of *Process* P_i or the granularity at which input data is used. Obviously, changing these parameters will change the resulting C_i (quality, certainty factor, etc.). However such alterations are done indirectly and the *Capability* C_i is fundamentally the same, i.e. it will serve the same purpose. A direct control over the effective C_i is achieved through the set of *Input Arguments*, which is the information required for the generation of *Capability* C_i.

It is also worth mentioning that the values of $[CA_i]$ will have impact on the *Resources* required for the corresponding *Process* P_i execution. If we change $[CA_i]$ in order to increase the quality of the resulting *Capability* C_i, it seems natural that more *Resources* will be required to compute it. Therefore, conveniently setting $[CA_i]$ will allow us to adapt *Processes* to the *Resources* available at each instant.

4.4.2. Modeling Emotion as Information

In Chapter 2 we have described how Emotions may be advantageously used as sources of information, in order to complete or simplify data gathered by perception (or previously stored in memory). We shall now model this interaction.

Let P_i be a *Process* related to *Capability* C_i. Let $[IA_i]$ be the set of *Input Arguments* of *Process* P_i. Then, $[IA_i]$ results from data belonging to:

- the *Model of the Environment*, [E(t)]
- Agent's *Proprio-Perception*, [P(t)]
- the *Output Arguments* of other *Processes*, $[OA'_i]$
- Agent's *Emotional State*, [Em(t)]

Alternatively,

$$[IA_i] = \{[E_i(t)], [P(t)], [OA'_i], [Em_i(t)]\}$$

where:

$$[E_i(t)] \in [E(t)]$$

$$[Em_i(t)] \in [Em(t)]$$

$$\forall\ [OA_j] \in [OA'_i], \exists\ P_j, \exists\ [IA_j], \exists\ [CA]_j, \exists\ C_j : P_j([IA_j], [CA]_j) \text{ -> } \{[OA]_j, C_j\}$$

The key point in the previous formula is the inclusion of the Agent's *Emotional State* as input to the *Process*, in much the same way as data from the *Model of the Environment* or from *Proprio-Perception*. As mentioned before, the information about the state of the *Emotional State* is stored in the *Emotional Accumulators* that are dynamically updated.

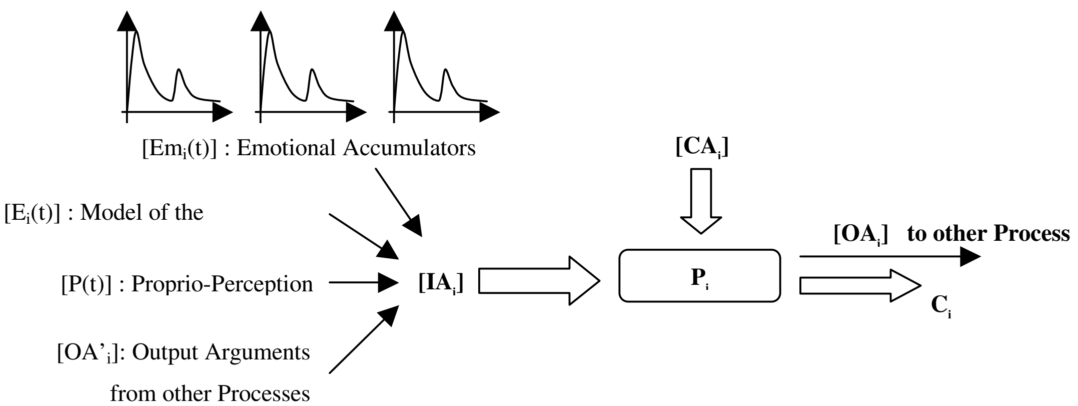

Figure 4.7. Emotion as Information. Values stored in Emotional Accumulators are included in the set of Input Arguments of Process P_i.

4.4.3. Modeling Process Control

We have also seen in Chapter 2 how Emotion helps to modulate each individual process by controlling several of its information processing parameters. Following our previous modeling effort regarding Agent's *Processes*, we may now model this type of interactions.

Let P_i be a *Process* related to *Capability* C_i. Let $[CA_i]$ be the set of *Control Arguments* of *Process* P_i. Then, $[CA_i]$ will be composed by data belonging to:

- the *Model of the Environment*, [E(t)]
- Agent's *Proprio-Perception*, [P(t)]
- a *Global Allocation Function*, A
- Agent's *Emotional State*, [Em(t)]

Again, alternatively,

$[IA_i] = \{[E_i(t)], [P(t)], A, [Em_i(t)]\}$

where

$[E_i(t)] \in [E(t)]$

$[Em_i(t)] \in [Em(t)]$

We have now included the influence of the Agent's *Emotional State* in the *Set of Control Arguments* of *Process* Pi. This can be done in two ways:

1. directly, by including the *Emotional State* [Em(t)] itself as a *Control Argument*
2. indirectly, by including the perception that the Agent has of its own *Emotional State* (included in Agent's *Proprio-Perception*).

The following picture illustrates this double influence of Emotional Mechanisms on the control of individual Processes.

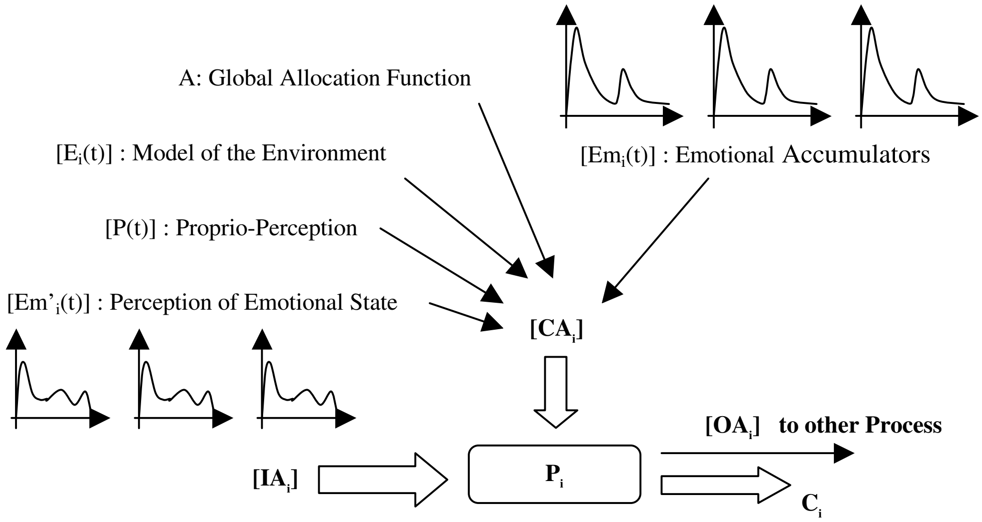

Figure 4.8. Emotion as a Process Control Mechanism. Both the values stored in Emotional Accumulators and Agent's Perception about its own Emotional State are included in the set of Control Arguments of Process P_i.

4.4.4. Modeling Resource Allocation

In the model that we have been presenting so far, resource allocation is accomplished by a *Global Allocation Function* that assigns Resources to specific *Capabilities*, thereby generating a set of *Effective Capabilities* [EC(t)]:

[EC (t)] = A([G(t)], [C(t)], [R(t)])

However, as we have seen, Emotions play an important role on global resource allocation. In order to model such interactions, let us simply extend our previous definition of *Global Allocation Function* to include the Emotional State, [Em(t)], as one of the parameters:

[EC (t)] = A([G(t)], [C(t)], [R(t)], [Em(t)])

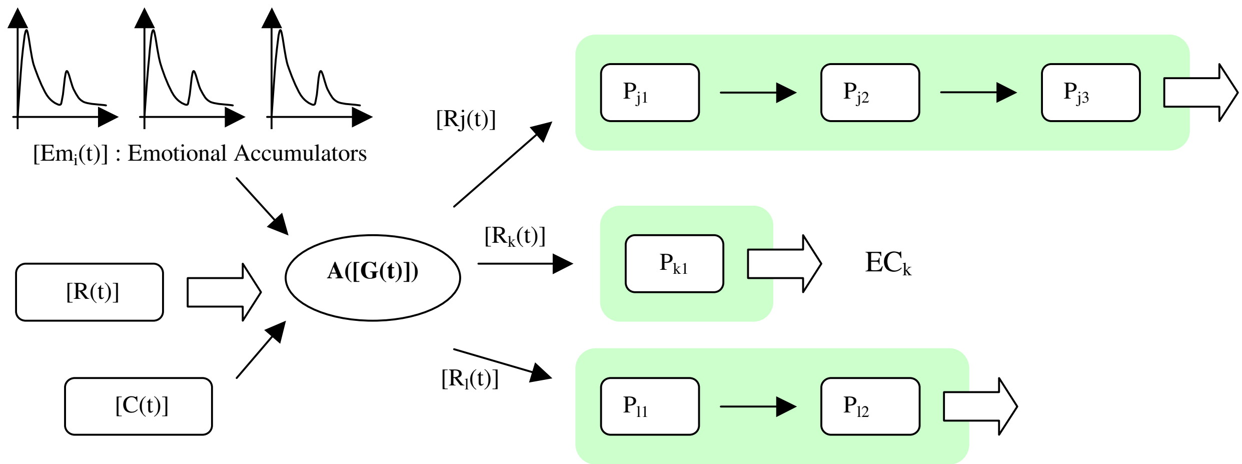

Figure 4.9. Values contained in Emotional Accumulators influence the way Global Allocation Functions assign resources to different Capabilities, which involve multiple processes

Once again we are reinforcing the feedback loop around EEF's with one more paths. Recall that one of the arguments of EEF's was the Agent's *Internal State*, in which we may find the set of *Effective Capabilities*. Furthermore, there is still another less direct loop that is established but whose importance must not be overlooked. In fact, the allocation of Resources to *Capabilities* will have a decisive impact on the Agent's success in achieving its Goals. Since the *Set of Goals* is an input of EEF's, another feedback loop is thus established.

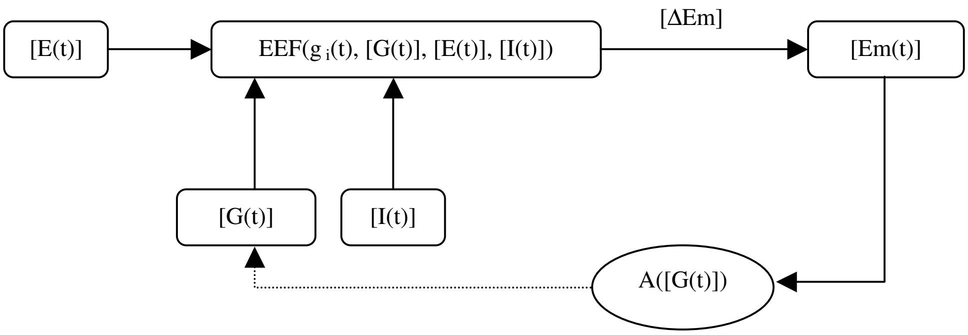

Figure 4.10. Another feedback loop is established through the influence of the Global Allocation Function, A([G(t)]), on the state of the Goals composing the Set of Goals, [G(t)]

4.4.5. Modeling Processing Strategies

In Chapter 2 we have defined four basic Information Processing Strategies that showed different patterns regarding the amount of Emotional Information used as well as the computational resources employed:

- *Direct Access*
- *Motivated Processing*
- *Heuristic Processing*
- *Substantive Processing*

When an Agent is operating under a specific Processing Strategy, all its processes tend to make a typical use of Emotion as information. Also, the control parameters of each individual process are set differently, resulting in distinct degrees of information processing complexity and, consequently, varying levels of the computational resources employed. At the same time, since computational resources are not infinite, a global scope mechanism is used to allocate the (limited) existing resources among capabilities, especially to those addressing more relevant Goals. As described in Chapter 2, Emotions play an important role at these three levels.

Therefore, using the previous three models we may also model the concepts related to Information Processing Strategies and its interactions with Emotional Mechanisms. We will need to combine the models of:

- Emotion as Information
- Process Control
- Resource Allocation

The next picture combines ideas from the three afore-mentioned models and presents a global perspective of our view of Emotional interactions within an Agent Architecture:

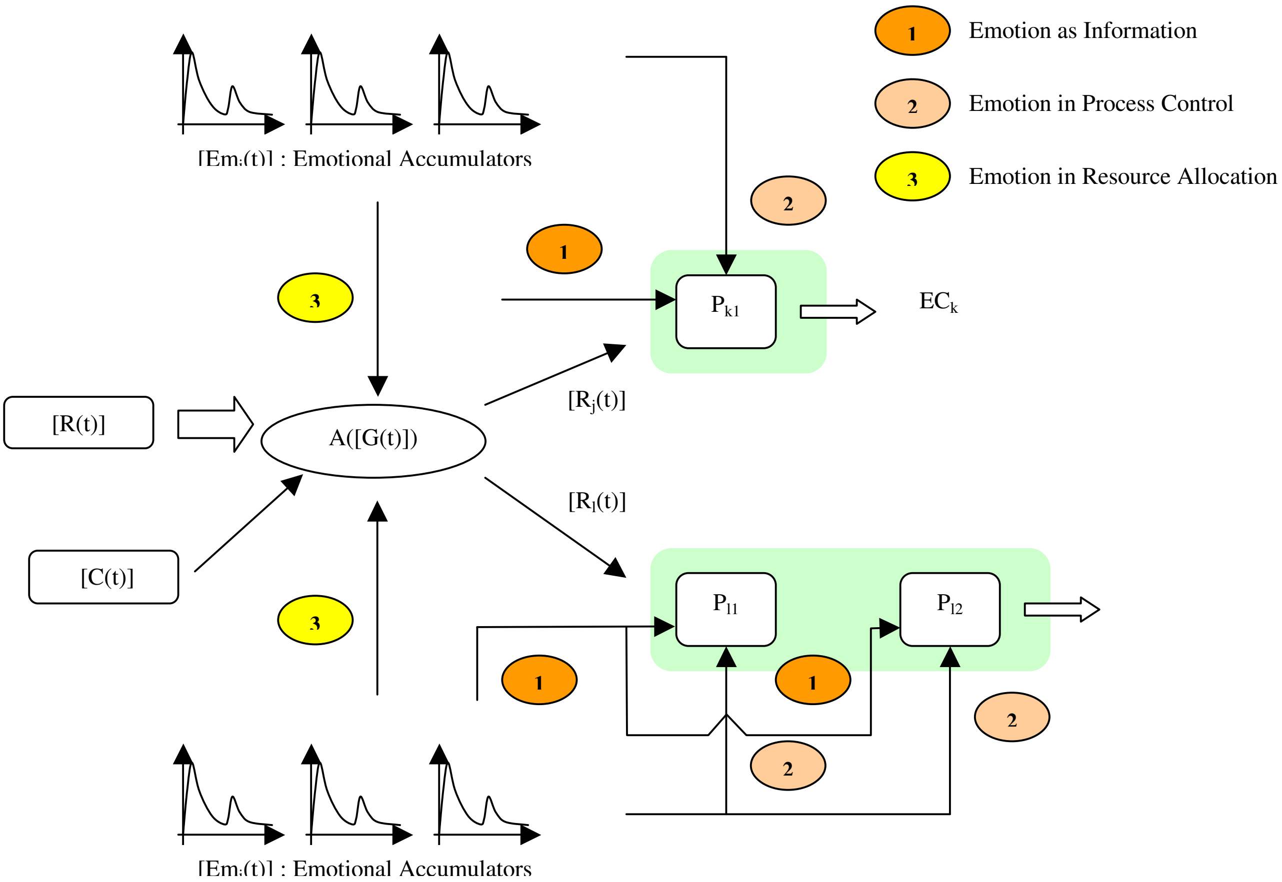

Figure 4.11.The overall influence of Emotional Mechanisms over Cognitive Processes (*Emotion as Information*, *Emotion in Process Control* and *Emotion in Resource Allocation*) allowing different global *Information Processing Strategies*

4.4.6. Emotional Agent

After all this modeling effort, we are now in position to complete the definition of Emotional Agent presented in Chapter 2. Firstly, let us restate the notion that Emotional Agents exist in complex[8] environments that require multiple and sophisticated *Capabilities* in order to achieve various (simultaneous) Goals. Therefore, an Emotional Agent is an Agent with complex *Capabilities*, supported by an Architecture that will eventually have multiple operating subsystems. Each of these

[8] The notion of *complex environment* we are considering has been explained in Chapter 2

Capabilities, and consequently the underlying operating subsystems, should possess several parameters that control its information processing methods.

Definition:

An *Emotional Agent* is an Agent that is able to operate under different *Emotional States* that are reached through a process of *Evaluation*. Such *Evaluation* takes into account the *Internal State of the Agent*, the *Environment Model* and the chances of achievement of each individual *Goal* belonging to the Agent's *Set of Goals*. The *Emotional State* may be used by other *Processes* within the Architecture (i) as input information, (ii) as a *Process Control* mechanism, (iii) as a *Resource Allocation* mechanism, and (iv) in the definition of the global information *Processing Strategy*.

4.5. Summary and Conclusions

In this Chapter we analyzed the global operating and architectural contexts for Agents intended for complex environments. Next, we defined and modeled several concepts regarding Emotion and Emotion-Cognition interactions. The key concepts discussed were:

- Emotional Elicitation
- Emotional Evaluation Functions
- Emotional State
- Emotional Accumulators
- *Basic Emotional Structures*

We then described how these concepts could be applied in modeling Emotion-Cognition interactions. We presented models for the following paradigms described in Chapter 2:

- Emotion as Information
- Emotion as a Process Control Mechanism
- Emotion and Resource Allocation Mechanisms
- Emotion and Information Processing Strategies

We finalized this Chapter with an extended definition of Emotional Agent.

CHAPTER 5. THE PYROSIM PLATFORM

5.1. Introduction

In this Chapter we will describe the Pyrosim platform that we have been building to support the development of our Emotion-based Agent Architecture. The Pyrosim platform has provided the basis for many experiments and has been very useful in clarifying our ideas about Emotion.

The Pyrosim platform simulates a forest environment where a team of Agents is placed to fight an ongoing fire. Each individual Agent can combat fire cells with a water jet that is connected to a limited capacity water tank. Agent mobility is constrained by its own simulated physical capacities (energy and acceleration) as well as by limitations imposed by characteristics of both the terrain and the fire. Fire propagation through the terrain, depends on the vegetation type and density, the terrain slope, and the wind, based on a realistic model. The Pyrosim also allows Agents to communicate in order to enable team efforts.

It may seem awkward to describe the platform before even summarizing the Architecture we are proposing, but we think this will help to explain the Architecture's underlying concepts. Although our Agent Architecture is not dependent on the Pyrosim Platform, some of its design requirements become more evident when focusing on this particular scenario. In fact, the Pyrosim Platform and our Agent Architecture have grown together. When we were developing our first Agents, we immediately identified specific points for improvements in the platform itself.

Throughout the next sections, we will be focusing on the platform's most relevant issues for Agent development. Although we will sometimes be forced to go into some detail about the implementation, we will try to describe the platform as much as possible in high-level. More detailed information about the platform and Agent programming is available in the "Pyrosim Agent Developer Manual", which is distributed with the platform. Issues about the mathematical modeling behind the simulation are described in Appendix A.

5.2. Motivation for Pyrosim

Our first attempts to build an Emotion-based Agent Architecture were made using a simple simulator named RealTimeBattle (RTB)[Ouc02]. This platform provides a simulated real-time environment where softbots fight for survival in 2D scenarios. RTB allows the developer to program their own softbots in C/C++, as well as to create custom 2D scenarios. Simple physical properties (air resistance, friction, material hardness) are also implemented to enrich the simulation. Softbots perception is basically a set of radar events from which they can detect walls, other softbots, shots and randomly distributed energy sources and mines. Softbots can accelerate, break, rotate and shoot in a given direction. The experiments made using RealTimeBattle allowed us to develop a sharper understanding of emotional-like structures, namely Emotional Evaluation Functions coupled with Emotional Accumulators [0li02].

However, we realized that to effectively test other emotion-cognition interactions, we would need a more complex simulated environment that would impose higher demands on the Agent. Simple environments do not reveal the need for emotional-like mechanisms. In order to be an appropriate test-bed for Emotion-based Agent Architectures, this simulation environment must meet the following requirements:

- high complexity for the participating Agent
- real-time requirements
- multiple concerns at stake
- autonomous decision-making coupled with multi-agent interaction
- closeness to a real-world problem[9]

These requirements are similar to those Humans meet when dealing with the real-world. As we explained in Chapter 2, Emotions may play a useful role in such a context because they help resource-bounded Agents to deal with complexity and to achieve multiple simultaneous goals. Simpler environments or Agents with fewer goals will simply not need Emotional mechanisms because possible problems may be solved with the help of simpler, more straight-forward mechanisms.

After trying to find a suitable environment simulation to test Emotional-based Architectures, we chose to build one from scratch. In our search for alternative simulation environments, most of the systems complex enough for Emotion-based Agents were based on a domain that we did not find appealing. Some of them were simulations of war, while others were based on toy environments.

Since forest fires are a major concern in Portugal, we decided to develop a forest fire simulation environment. We knew that building a simulation environment was a considerable effort. However,

[9] This is not really a requirement but any simulation environment that complies with this condition will almost certainly ensure the previous four conditions.

such a platform would also be useful for other studies in the field of autonomous Agents such as, for example, in multi-agent coordination. Furthermore, being able to control some of the parameters of the simulation environment would also be advantageous for experimenting new features in our Agents. The balance between effort and possible future advantages seemed positive enough to start Pyrosim's development.

5.3. Overview of the Pyrosim Platform

The Pyrosim Platform is a simulation environment composed by a set of applications and modules that communicate through a network infrastructure. The platform was designed to run as distributed as possible in order to allow a balanced scaling of the simulation. Each module of the platform can be run on a different computer to avoid heavy computational bottlenecks.

The core of the Pyrosim Platform includes two basic elements:

1. the *Pyrosim Server* application, responsible for running all the simulation logic. It contains the environment's model and updates the state of every entity in each simulation cycle.
2. the *Agent Skeleton* layer that provides a convenient interface for creating customized Agents. It manages all low level communication between a customized Agent and the Pyrosim Server.

There is also another application that is very useful for monitoring the state of the simulation: the Pyroviz. The Pyroviz allows the developer to visualize almost in real-time what is happening during the entire simulation through a 3D rendered scene, as well as track all the communication activity between the participating Agents.

Figure 1 illustrates the relationship between the three components of the Pyrosim Platform and lists some of their specific functionalities.

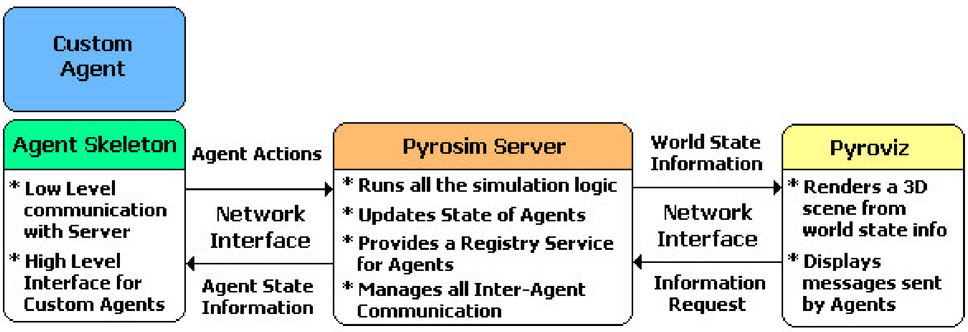

Figure 5.1 - Overview of the Pyrosim Platform

5.3.1. The Pyrosim Server

The Pyrosim Server is the main application of the Pyrosim platform: it is inside the Pyrosim Server that all the simulation is run. The Pyrosim Server supports the entire environment model. It manages all Agent interactions, vegetation elements, fire spots, temperature, terrain and wind.

The Pyrosim server runs continuously in cycles throughout the simulation. In each cycle the server updates the state of all entities under simulation, a process that requires a reasonable amount of computation effort, especially when the environment is large and has a numerous amount of entities.

The Pyrosim server sets up an Agent Registry Service that accepts network connections from Agents, which connect through their Agent Skeleton layer. After a successful registration process, the server feeds networked Agents with updated state information and receives their commands to act upon the environment. All communication established between the participating Agents is also supported by the Pyrosim Server that acts as a special message router.

In order to allow Agent developers to understand what is happening during the simulation, the Pyrosim Server supports network connections from a dedicated visualization application, the Pyroviz. This application requests information about the state of the environment and renders a 3D scene to help users visualize what is happening. This inspection facility provided by the server may be used by other applications that may use the information from the simulation to create other interfaces with the user (2D, speech, etc.) or to process it for different purposes (logs).

It is worth mentioning that there are several parameters (e.g.: terrain geometry, vegetation density, etc.) of the simulation that can be defined using a special configuration file, making it possible to simulate a wide variety of situations.

5.3.2. The Agent Skeleton

The Agent Skeleton is a software layer that is the basis for Agent development. The Agent Skeleton provides developers with several essential features that isolate programming from the platform details. In fact, we think the name Agent Skeleton is quite appropriate for the role it has in custom Agent programming. The Agent Skeleton handles all low-level communication with the Pyrosim server and offers the developer a high-level interface with several methods to consult the state of the Agent and to act upon the environment.

One of the Agent Skeleton's major tasks is to provide updated information about the Agent's state without requiring explicit intervention from the developer: the updating process is performed automatically. The Agent Skeleton is constantly requesting the server for updated information about the Agent's state, including several parameters, such as:

- the physical energy;
- the location, moving direction and slope;

- the speed and acceleration;
- the status of the personal water jet;
- visual maps of the entities perceived by the Agent.

Before being able to participate in the simulation each Agent has to register in the Pyrosim server under a unique name. The Agent Skeleton also provides a simple mechanism intended to interact with the server during the registration process, releasing the developer from the burden of directly negotiating the name of the Agent with the server.

Finally, the Agent Skeleton also provides the developer with high-level methods that enable Agents to exchange messages between them. The Agent Skeleton supplies an interface to send broadcast or private messages to other Agents. On the other hand, in order to assist the developer in handling the messages the Agent receives, the Agent Skeleton layer implements a custom event mechanism that reacts to the arrival of new messages.

5.3.3. Pyroviz

The Pyroviz is a visualizer application that creates 3D views of the environment and updates it automatically throughout the simulation. The Pyroviz establishes a network connection with the Pyrosim server and requests for descriptive information about the environment:

- geometry of the terrain
- position of the vegetation elements
- position and action of the Agents
- position and intensity of the fire spots
- …

Using all this information, Pyroviz renders a 3D scene of the environment and, after completing the rendering process, it makes a new request to the server for updated information. The rendering process may consume a significant amount of processing power. Therefore, the Pyroviz's refresh rate will heavily depend on the 3D graphic capabilities of the computer where it is running.

The Pyroviz will also display all messages exchanged by the Agents in order to allow a complete monitoring of the communication that takes place between Agents.

5.4. The Pyrosim World

In this section we will describe the simulated environment, which we will call Pyrosim World. This will be a high-level description, focused mainly on presenting the important details for Agent development. We will not go into details about the mathematical model that supports the simulation as it will be covered in Appendix A.

The Pyrosim World is built upon a grid of NxN cells, named Pyrosim cells. In each simulation cycle the state of all simulation cells in the World must be updated. The dimension of the Pyrosim World is set by default to 128 x 128 cells, which amounts to a total 16384 cells. The physical dimension of a Pyrosim cell is 16 meters. Therefore, the default physical dimension of the World is 2048 m x 2048 m. This dimension should be taken into account when programming the Agent: walking or running for 2Km may take a considerable amount of time and physical energy.

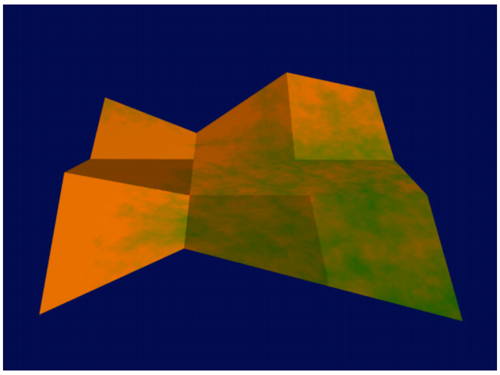

Figure 5.2 – A 3x3 sample of terrain.

A Pyrosim Cell is considered to be homogenous, so that all properties are constant along the cell. The next figure illustrates both the structure and main properties of a Pyrosim cell.

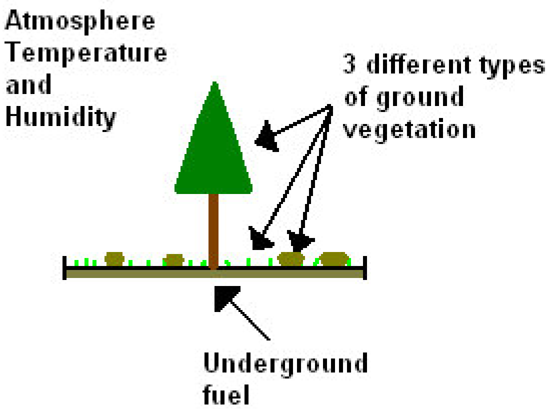

Figure 5.3 - A simplified view of a Pyrosim cell.

Each simulation cell may be populated by three different types of ground vegetation (grass, bushes and trees) and by underground vegetation. These four vegetation elements have distinct values for three important properties concerning fire resistance and combustion:

1. Temperature of Ignition – the temperature at which the vegetation element will start burning: T_{MAX} (ºC).

2. Burning Rate – the amount of heat produced in each second by the burning element: BR (kJ/s).
3. Total Energy – the overall energy that a vegetation element has to burn initially: E_{TOTAL} (J).

These three properties have a decisive influence on the fire dynamics and should be well understood in order to predict the fire progression in the terrain. The next table presents a comparison between the four types of vegetation regarding temperature of ignition (T_{MAX}), burning rate (BR) and Total Energy (E_{TOTAL}):

	T_{MAX}	BR	E_{TOTAL}
Jndergrounc	Medium	Low	Medium
Gras	Low	High	Low
Bushe	Medium	Medium	Medium
Tree	High	Medium	High

Table 5.1 - Comparison between the 4 types of vegetation regarding the properties that influence fire dynamics (based on common-sense knowledge).

5.4.1. The Process of Ignition

In Pyrosim, there are two ways of igniting a vegetation element. The first one is by deliberately starting a fire in a given cell. This operation can only be done by the Pyrosim Server and is usually performed in a random spot at the beginning of the simulation.

The other way results from the progression of fire itself during the simulation. When a given region in the World is burning, a percentage of the heat produced during the combustion is transferred to the atmosphere, and another percentage is transferred to the neighbor cells. The next figure (3) illustrates the heat transfer process among neighbor cells.

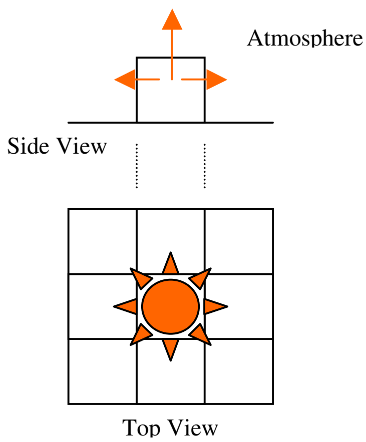

Figure 5.4 - The Heat transfer between cells. The combustion in the central cell is producing heat that is transferred to the atmosphere and to the neighbor cells.

If the temperature of any of the neighbor cells eventually reaches the temperature of ignition (T_{MAX}) of a vegetation element in that cell, that vegetation element will start burning. Since this recently ignited element will also produce heat, the chances of other vegetation elements in the same cell starting to burn will increase significantly. This process results in sustained fire propagation to more cells due to the increase in the amount of heat produced and consequent heat transfer to neighbor cells.

5.4.2. The Influence of Terrain Geometry

It is a well-known fact that fire tends to propagate faster upwards. This is because the heat air masses produced by an ongoing combustion are lifted and therefore transfer most of the heat to the material above the combustion.

Pyrosim considers this effect when calculating the heat transference between adjacent cells. The amount of heat transferred depends on the slope of the terrain, increasing in the direction of the most elevated cell. This results in faster fire propagation along ascending paths, with fire climbing up hills faster than it proceeds over flat regions (S1 > S2).

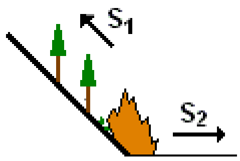

Figure 5.5 - The speed of fire climbing up the hill (S1) is higher than the speed of progression along a flat region (S2).

5.4.3. The Influence of Wind

Wind has also a strong influence on the heat transfer between cells. Depending on its speed, wind will move a varying amount of heat in a given direction, reinforcing or diminishing the heat transfer that would naturally take place between two adjacent cells.

For wind speeds exceeding 50 Km/h, the impact of wind on fire propagation is very significant. By incrementing significantly the transfer of heat in its direction, wind will make the fire propagate faster in that direction. In such a situation, the fire-front usually becomes much larger, opening up widely. On the other hand, fire will progress very slowly against the wind. Figure 5 illustrates this case.

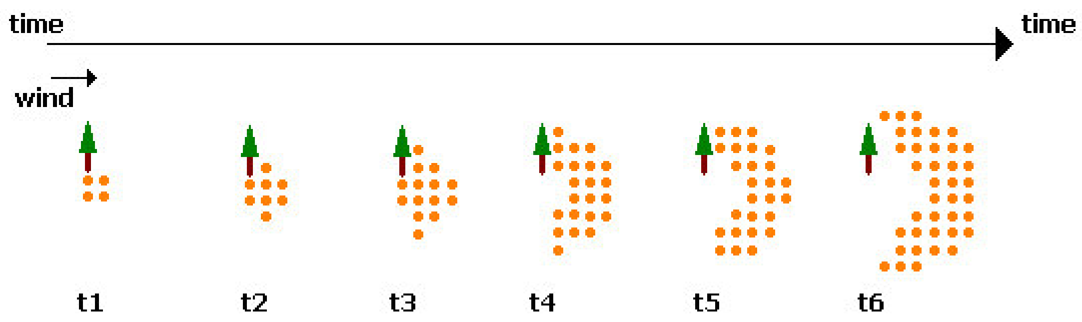

Figure 5.6 - The effect of strong wind on the evolution of the fire-front. Notice the widening of the fire-front during the fast progression of fire in the wind direction.

When the wind changes direction very quickly, the propagation of fire becomes very unstable and difficult to predict. Rapidly changing wind causes extremely complex and dangerous fires.

5.4.4. Fire Extinction

The process of fire extinction is basically the inverse of that of ignition. Extinction happens naturally whenever the temperature around the flame drops below a given critical point. Below that temperature, combustion is not sustainable and fire will extinguish.

When fighting fire with water, as our Agents do, the extinction of a combusting element is achieved as a result of the sudden temperature drop around the flame, caused by water evaporation. Water has the capability of absorbing great amounts of heat, which is then dissipated by evaporation. The water thrown over the burning element absorbs the local heat, evaporates and reduces the temperature around the flame. Eventually, the temperature will drop below the critical point and the flame will fade out.

Agents may throw water at burning cells using a personal water jet. Agents may control the following variables of the water jet:

- the flow of water, i.e., the volume of water thrown per unit of time;
- the distance that the water is being ejected;
- the direction of the water jet.

The Agent only has direct control over water flow and jet direction. The distance at which the water is being ejected is limited by the maximum power of the water jet and, therefore, depends on the flow of water being ejected. If the Agent demands a very high water flow from the jet, then it may not be able to project the water as far as the Agent intends. To reach the intended distance, the Agent may have to reduce the flow of water being ejected.

5.5. Pyrosim Agents

As we have mentioned before, the Pyrosim Platform provides a software layer, the Agent Skeleton, which helps developers to build higher-level Agents. The Agent Skeleton is not in itself an Agent, it does not have any behavior. However, it provides a basic set of capabilities for perceiving and acting on the Pyrosim World that will be described throughout this section.

It is also important to refer that the Agent Skeleton does not possess any intrinsic “emotional” capability: all the Emotional mechanisms were implemented on top of this layer, as will be shown in Chapter 6.

In this section we will focus on the following issues concerning the Pyrosim Agents and Agent Skeleton:

- Proprio-perception
- Visual Perception
- Action Set
- Communication

For further details about the Agent’s programming, please refer to the Pyrosim Developer Manual, which is distributed with the Pyrosim Platform.

5.5.1. Proprio-perception

The most elementary data provided by the Agent Skeleton is the state of the Agent, which comprises several parameters intrinsically perceivable. These parameters can be considered the Agent’s proprio-perception since they are always known and no special query action needs to be executed in order to find its value.

The Agent’s proprio-perception includes the following (read-only) parameters accessible through the Agent Skeleton:

- X and Y coordinates of the Agent’s location. Range depends on the simulated terrain dimensions (default range: 0 to 128).

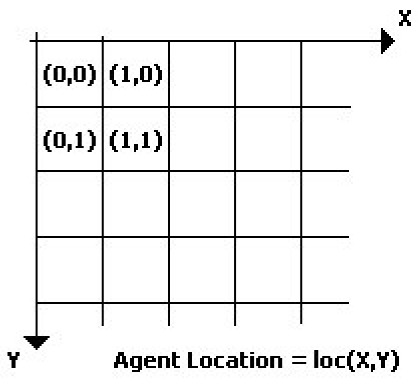

Figure 5.7 – Pyrosim coordinate system

- height (Z coordinate) of the Agent. Range depends on the terrain dimensions but is always a positive value.
- energy level of Agent, ranging from 0 to 100.
- skin temperature of the Agent in degrees Celsius. Values over 50º will result in damage to the Agent and in fast decrease of the energy level.
- rotation in degrees of the Agent's direction, measured clockwise in the XY plane.

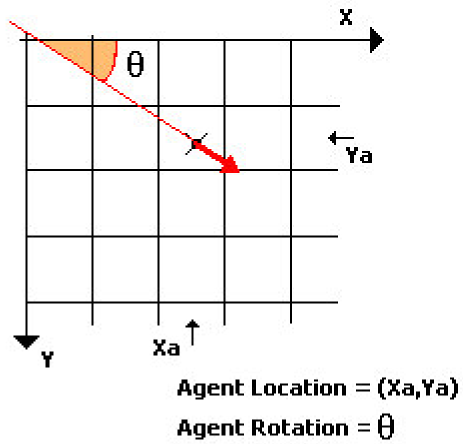

Figure 5.8 – The rotation angle θ is measured clockwise from the X axis

- the tangential acceleration the Agent is trying to produce. This value ranges from –1.73 m/s^2 to 6.92 m/s^2. The 6.92 m/s^2 value corresponds to the acceleration needed to climb a 45º slope terrain. This means that the Agent will not be able to proceed forward when the slope of the terrain is higher than 45º.
- the speed in m/s along the X axis and along Y axis.
- the slope of the terrain along the X axis and Y axis.

- The Agent Skeleton also provides access to information about the Agent's water jet:

- operating state of the water (on or off).

- the distance the Agent intends to reach with the water jet. Depending on the water flow employed, the desired distance may be reached or not. In practice, it corresponds to controlling both the orientation and the flow of the water jet. The maximum absolute value is 40 meters.
- the flow in lt/s that the Agent intends to eject. The maximum absolute value is 16 lt/s.
- the distance the water jet is effectively reaching. The Agent does not control this value.
- the effective water flow of the water jet.
- the angle made between the water jet direction and the forward direction of the Agent.
- X and Y coordinates of the cell being targeted by the water jet. The Agent does not have direct control of this value.
- the level of the tank in lts.

5.5.2. Visual Perception

The Agent Skeleton also provides information about what the Agent would see in its surroundings. This information is not an image, as if it was captured by the Agent's eyes, but a structured description of the features that would be extracted from such an image. Therefore, we may consider this information as the visual perception of the Agent.

Visual perception is organized into several 2D maps called Float Maps. These maps store the properties of a given element in the environment. Changes in the environment may be detected by comparing two consecutive Float Maps. The Agent Skeleton has several Float Maps available that numerically describe the properties of the surroundings:

- terrain geometry;
- type of vegetation;
- intensity of fire;
- level of damage caused by fire;

For example, the Float Map concerning fire intensity would store the values of fire intensity in each of the cells around the Agent. A value of 0 would mean that the given cell is not burning and a positive value would describe the intensity of the ongoing fire.

In order to account for the expected loss of detail of more distant objects two specific Float Maps are supplied for each of the categories listed before:

1. a Close Range Map;
2. a Medium Range Map;

Close Range Float Maps are related with the Agent's immediate surroundings and, therefore, describe the properties of a given category with a higher level of detail. Each Close Range Float Map consists of a 3x3 array of float values describing a certain property both of the terrain cell where the Agent is located and of the eight surrounding terrain cells. The next sketch shows the 3x3 cell area covered by a Close Range Map (the Agent is located in the center cell).

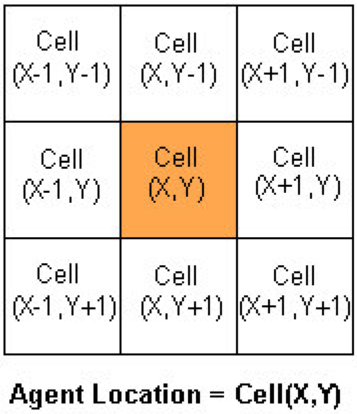

Figure 5.9 – The cells comprised by a Close Range Map.

On the other hand, the purpose of a Medium Range Float Map is to provide a description of a much wider (and distant) area, namely 21 x 21 terrain cells. However, the increased area coverage is obtained at the cost of detail. In fact, instead of being composed of 21 x 21 float values (one for each terrain cell covered), a Medium Range Float Map is composed of just 7x7 elements: each element of a Medium Range Float Map describes the *average* value of property over a 3x3 cell area. The next picture tries to describe these correspondences.

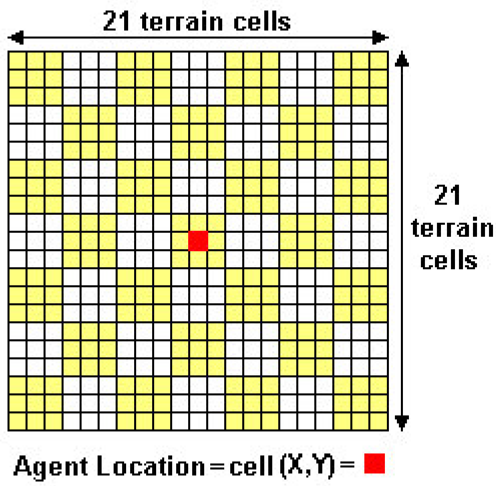

Figure 5.10 – The cells inside a Medium Range Float Map. The Medium Range Float Map stores a 7x7 grid of the average values of the 3x3 checkered areas.

As illustrated above, the 7 x 7 checkered pattern covers an area of 21 x 21 terrain cells. The Agent is located at the center of this 21 x 21 area. This particular terrain cell and the eight terrain cells surrounding it, comprise the Agent's close range. As we have seen before, a Close Range Map describes these nine cells in much more detail.

However, all Float maps are subjected to noise, altering the numerical descriptions they store. The further away the cell is from the Agent, the more significant will be the influence of noise distortion.

5.5.3. Visual Occlusion

Pyrosim server takes into account the possibility of occlusion during visual inspection. Whenever the geometry of the terrain (e.g. a hill) intersects with the Agent's line of vision to the center of a given terrain cell, the Agent will not be able to receive visual information about that cell. Figure 15 tries to explain this limitation imposed by the server:

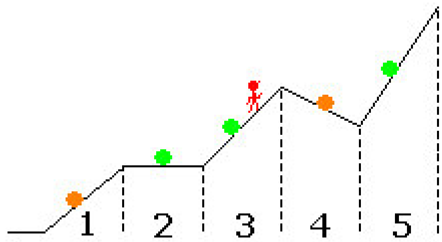

Figure 5.11 – The fireman located in cell 3 will be able to receive information about cells 2, 3 and 5. Both cell 1 and cell 4 are occluded by the geometry of the terrain.

For all Close Range Float Maps, occlusion is ignored to simplify the task of the Agent at a local stage. However, for Medium Range Float Maps, the server checks how many of the nine cells that compose each 3 x 3 cell area covered are occluded. If more than four cells are in fact occluded, then, the whole 3 x 3 area is also considered occluded. In this case, the value stored in the Medium Range Float Map to describe the occluded 3 x 3 area will be –1.

Note that none of the Medium Range Float Maps presented so far can store negative values to describe the corresponding property, because all properties have always non-negative values (even the height of the terrain). A negative value will always represent occlusion.

5.5.4. Perception of other Agents

The Agent Skeleton also stores information about other Agents in the simulation. Our Agent is able to see all other Agents inside the Medium Range Area (21 x 21 cells) if these Agents are not occluded, as described in the previous section.

The information gathered about other Agents is:

- the name of the Agent;

- the X coordinate of Agent location;
- the Y coordinate of Agent location;
- an estimate of the current energy of the Agent;
- a description of the current action of the agent;

5.5.5. Action Set

The Agent Skeleton layer has several methods available for Custom Agents to act upon the World. At the Agent Skeleton level, the set of possible actions comprises only basic low-level actions for movement and for control of the water-jet. More complex tasks, (e.g. walking to a target point), are not considered *basic low-level actions* as they may involve multiple criterion decisions. Therefore, this type of actions should be implemented in higher-level classes.

Controlling moves

Agents may control two basic parameters of their movement: the acceleration effort and the direction of movement.

Agents are able to produce positive acceleration efforts up to 6.92 m/s^2, which is equivalent to the acceleration effort needed to climb a 45º slope limit. Therefore, Agents will not be able to climb hills with slopes higher than 45 degrees, unless they accelerate sufficiently before starting to climb the hill. In addition, Agents will not be able to stop their movement when descending hills with slopes higher than 45 degrees, which represents a risk to avoid.

Backward (negative) acceleration is also limited to 1.73m/s^2 (25 % of the maximum positive acceleration), which results in Agents not being capable of moving backwards as fast as they move forward.

Controlling Water Jet

Agents can control four parameters of their water jet:

1. the On/Off switch;
2. the intended distance at which the water is ejected (absolute limit 40 meters);
3. the direction of the jet;
4. the flow of the jet in liters/s (absolute limit 16 liters/s).

However, since the water jet has limited power, there is a dependency between the flow of the jet and the distance at which the Agent intends to throw water: higher flow will reduce the possible distance range. The Agent may control the flow of the jet directly but he has no guarantee that the water will reach the intended distance. This implies that the Agent may have to deliberately reduce the flow of ejected water to ensure that the distance he wants to throw water at is actually reached.

However, this will decrease the volume of water that the Agent can throw in a fire cell. Therefore, in order to achieve maximum fire fighting efficiency, *the Agent has to be as close to fire as possible*, although this may become dangerous.

5.5.6. Communication

During the simulation, the Agent may send other Agents two different types of messages:

1. broadcast messages: messages sent simultaneously to all Agents. All Agents in the simulation will receive the sent message.
2. 1-to-1 messages: messages sent exclusively to one Agent. Only the selected target Agent will receive the message.

To optimize message processing, the Agent Skeleton has an internal event-driven mechanism that signals new incoming messages.

The platform does not constrain by the structure of the message content. Pyrosim simply delivers a string between two Agents whose structure is irrelevant for the message passing mechanisms. Any message protocol has to be implemented in the Agent-Skeleton.

5.6. Pyrosim and Emotional Agents

Let us now analyze Pyrosim in the perspective of the requirements that we considered essential for any environment where Emotions might be useful. We will be pointing out situations where Emotional-mechanisms may come into play.

High level of complexity for the participating Agent

Pyrosim is a complex environment. Using the taxonomy from [Rus95], the Pyrosim world is:

- *Partially Accessible*. Agents only have *partial perception* of the environment. They can only visualize accurately objects in their close surroundings (3x3 cells area). Information about objects located further away has an inferior level of detail and is altered by noise. Additionally, there is always the chance of visual occlusion. After a certain point, the Agent is not able to get any more "visual" information. Reports about the global state of the fire are available only from time to time and have a limited degree of detail and accuracy. *Agents may have to find a way to compensate for a possible absence of information.*

- *Non-Deterministic*. Although the world evolves in accordance to a defined set of rules, the number of unknown or uncertain parameters (terrain geometry, wind, vegetation) is large enough to make it very difficult to predict. Furthermore, the outcome of an action is not certain, at least in an indirect way. The Agent is not sure if pointing its water jet to a fire cell will extinguish it. Actions may be carried out according to a variable set of parameters that establish a compromise between performance and resource cost. *The Agent might require a method to adapt these parameters according to its resources.*
- *Dynamic*. The world is constantly changing, even if the Agent does not intervene. Fire-fronts progress in the terrain, wind speed and direction change, vegetation is destroyed. The Agent itself is always changing (e.g.: loosing energy, slipping from a hill). Other Agents in the environment will take action (change position, choose another target cell) and change the world according to their own criteria. *The Agent will need a mechanism capable of detecting the most relevant changes and situations where more attention should be given to environment analysis.*
- *Continuous*. This is, of course, a computer simulation run on a discrete machine but we may consider that for the Agents the simulated world is continuous. We may consider that the world may be in one of an infinite number of states. *Agents may need a mechanism to reduce the state dimension of the environment in order to make it more tractable.*

In order to act in such a complex world, the Agent will need multiple capabilities and those capabilities will naturally be dependent on several parameters. Managing all the capabilities with resource constraints is essential for Agent's success.

Real-time requirements

The environment demands real-time action from the Agent. The fire is not stopping. The world is both dynamic and dangerous for Agent Goals (e.g.: survival). The Agent needs to be constantly perceiving the environment and to act accordingly. However, there might not be enough time/resources to perform an exhaustive analysis to decide the "best" action to be taken. *Agents might need a mechanism to adapt their response time to environment requirements and a method to balance the amount of time spent on environment analysis and action control.*

Multiple concerns at stake

Each Agent has to ensure its own safety while fighting the fire. At the same time, the Agents need to help the team to remain safe, which implies that they comply to certain restrictions (remain close, remain in line-of view, etc). Additionally, certain points in the map may be considered vital (e.g.: houses or industrial installations) and the team has to ensure that fire will not damage them.

There are various simultaneous (sub)goals to be attained. *The Agent will need a mechanism to help him decide which is the most relevant or urgent goal at each moment.*

Multi-Agent System

No single fireman is able to put out a reasonably large fire by himself, so, there is an obvious need for cooperation. However, since the Agent has both individual and team goals to attain, a *balance between individual action and cooperation must be established*. Furthermore, decisions taken by a group leader affect the entire team and need to consider very complex team factors (previous performance, current state of the team), besides the threats imposed by the environment. *Besides autonomous decision-making and interaction capabilities, a robust mechanism that ensures stability to leadership decisions would be advantageous.*

Closeness to a real-world problem

This is not a requirement *per se* but closeness to a real world situation may help understand the role of Emotional-mechanisms. This does not mean that the Emotional mechanisms used should mimic those of Humans. However, since the real world is a very complex world that we know reasonably, we may use this knowledge to decide where Emotional-mechanisms should be employed more advantageously.

These properties point towards the need for Emotional Mechanisms, as those described in the previous Chapter. Naturally, Emotional Mechanisms are only a part of a more global and complex Architecture. Therefore, implementing an Agent with Emotional capabilities is by no means simpler than implementing an Agent with no Emotion-based concept present in its Architecture. An Emotional Agent possesses a regular set of functionalities that would be found in Agents of other type. Additionally, an Emotional Agent uses Emotional Mechanisms to interact, control or help other mechanisms in their usual functions. Emotional mechanisms are not intended to substitute other more standard functionalities but may be useful in improving their global performance.

In the next chapters we will describe our Architecture and explain how Emotional Mechanisms help to solve some of these issues.

5.7. Summary and Conclusions

In this chapter we presented the Pyrosim platform that has been especially built to support the development of Emotion-based Agent Architectures. We presented the motivation for building our own simulation platform instead of using other simulators available. Next, we made a brief overview of the Pyrosim platform and showed how it is composed by three elements: the Pyrosim simulation server, the Agent Skeleton layer and the Pyroviz visualizer.

We focused on the simulated environment and explained several factors that contributed to its complexity. We have also described the basic functionalities provided by the Agent Skeleton in order to build higher-level Agents, namely those functionalities regarding proprio-perception, “visual” perception, action and communication.

We proceeded to analyze why the Pyrosim environment is appropriate for building Emotional Agents. The environment’s main characteristics were enumerated and we pointed out some problems that suitable Agent Architectures will have to deal with in order to be successful in the Pyrosim environment. We finished making some suggestions about where Emotional mechanisms may be advantageously used to help solve those problems.

CHAPTER 6. AN EMOTION-BASED AGENT ARCHITECTURE

6.1. Introduction

In this Chapter we will describe the Emotion-based Architecture that we have developed. Although this Architecture has been instantiated and customized for the Pyrosim simulator (described in the previous Chapter), it may nevertheless be considered a general mentalist-like Agent Architecture, applicable in other domains. The concepts behind Emotional Mechanisms, which form the core of the Architecture, could easily be transposed to other application domains keeping their underlying implementation almost untouched.

The proposed Architecture is composed of two layers built on top of the basic Agent Skeleton layer, already provided by the Pyrosim platform. As we have seen in the previous Chapter, the *Agent Skeleton* layer is responsible for dealing with all low-level communication details with the simulator server and may be seen as the Agent's set of sensory and low-level motor capabilities.

The two layers built on top of the *Agent Skeleton* layer have several specific Goals associated. They also have a set of increasingly high-level capabilities closely related to the achievement of the corresponding Goals. Immediately above the *Agent Skeleton* layer, we have placed the Basic Control Agent Layer (BCA Layer). This layer is mainly concerned with the Agent's most important Goal, survival, and possesses all the basic Capabilities necessary to ensure its achievement. On top of the BCA Layer we have built an additional layer, the Basic Deliberative Agent Layer, which provides more complex Capabilities required for the Agent's practical Goal, fire-fighting.

Despite this conceptually layered disposition, our Architecture is composed of several distributed *Functional Modules* whose place within the layers is sometimes not so self-contained. Some *Functional Modules* are shared between the layers because they may operate at different levels of abstraction and have different customizable operating settings. In fact, the boundaries between layers are much more connected with how the processor time is allocated to each functional module than with anything else. Figure 1 tries to briefly illustrate this.

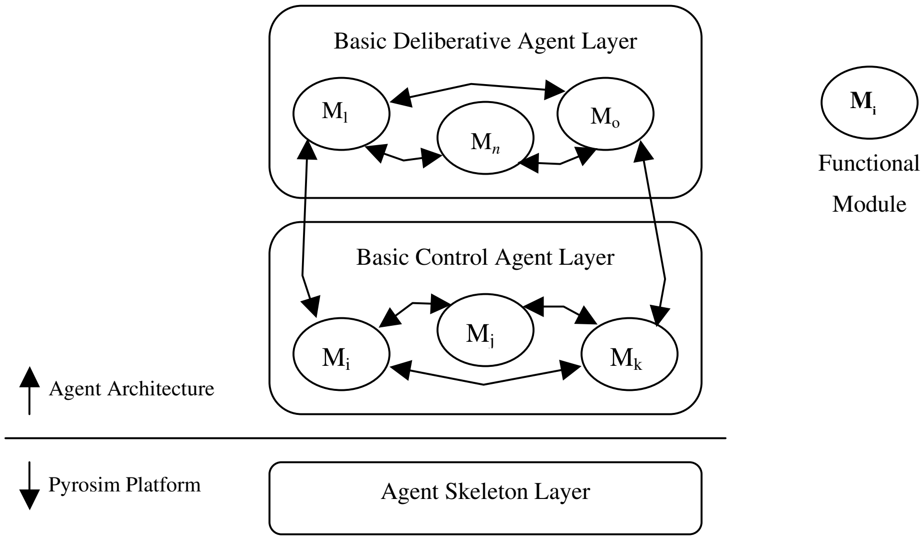

Figure 6.1. Overview of the Architecture: a layered disposition encompassing several Functional Modules

In the next sections of this Chapter, we will present the details of our Agent Architecture. We will to introduce the basic concepts and elements of our Architecture without focusing too much on Emotional interactions, which we will address in the next Chapter.

6.2. Functional Modules

Functional Modules are the building blocks of our Architecture. They are responsible for providing the Agent with the necessary Capabilities to act on the environment. Some of these Capabilities are connected with tasks or actions regarding a direct intervention in the environment (e.g.: moving, planning), while others are more concerned with the Agent's internal parameters (perception, management of internal resources) or with their execution. From a conceptual point of view, they may be grouped in four distinct categories:

1. ***Perceptors*** – examine low-level sensory data, provided by the *Agent Skeleton* (proprio-perception and visual maps), to produce higher-level perceptional *Beliefs* (e.g.: distances to closest fire cell, speed of fire-front propagation). *Perceptors* are mainly concerned with the evaluation of local data and are not usually very processor intensive.
2. ***Reasoners*** – these may be considered higher-level modules operating mostly on *Beliefs* produced by Perceptors. *Reasoners* are mainly concerned with decision-making (e.g.: deciding a new goal), planning (e.g.: path-planning), forecasting (e.g.: prevision of fire-front progression) as well as with high-level data analysis (e.g.: analysis of the global fire report). *Reasoners* usually demand a greater amount of processor time and have multiple parameters to change its operating modes.

3. **Managers** - managers are intended to work as schedulers and coordinators. Our Architecture includes one *Goal Manager*, responsible for allocating processor time to Goals, and one Action Manager that pipes in actions (agent locomotion, water jet control, etc.) to the Skeleton Layer. Managers are responsible for helping to detect and resolve possible conflicts, such as conflicting Goals or Actions.
4. **Controllers** – Controllers transform high-level commands into low-level actions (e.g.: rotation, acceleration, etc.) that are then sent to the Action Manager. These are not usually very processor intensive modules but, since they are closed-loop controllers, they need to be allocated frequently.

In practice, these categories correspond to a set of Java Interfaces and base classes from which specific *Functional Modules* are developed.

Besides these modules, the proposed Architecture also includes a memory structure that works similarly to a blackboard and that we named *Working Memory*. The purpose of the *Working Memory* is to allow other *Functional Modules* to exchange structured information (i.e. *Beliefs*) with each other. *Functional Modules* may use *Working Memory* to store and retrieve *Beliefs* indexed by keyword. For example, after processing local sensory data, *Perceptors* can store in *Working Memory* all new *Beliefs* produced, which then become available as input for *Reasoners* to produce supplementary higher-level *Beliefs* (e.g.: a plan, a global fire analysis). All these *Beliefs* can be associated with each other in order to build a Belief cluster available to other modules, if needed.

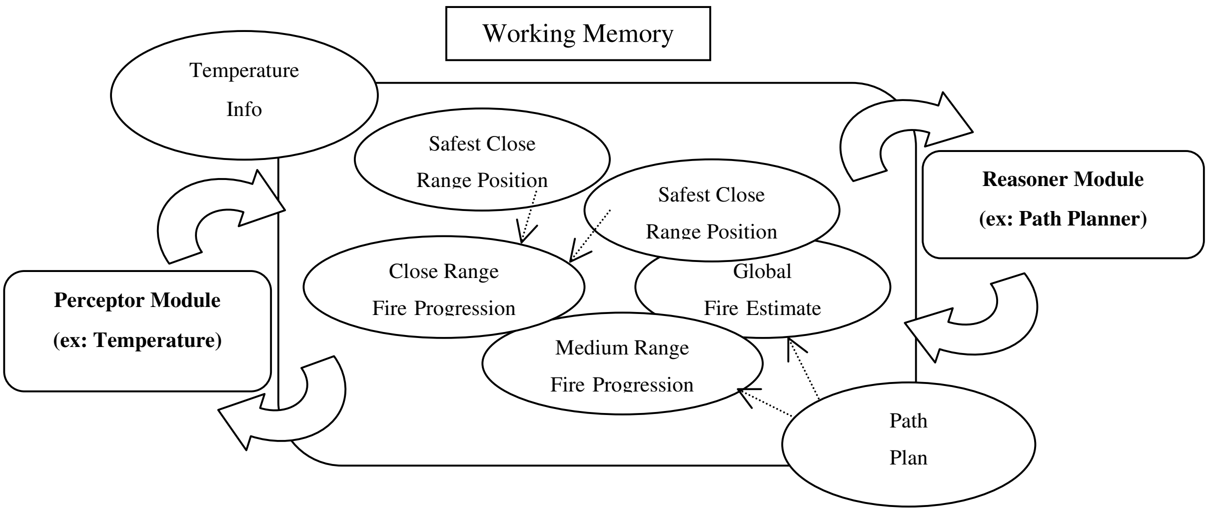

Figure 6.2. The Working Memory works as a repository of Beliefs produced by Functional Modules. Additionally, all Functional Modules may freely access the Working Memory to obtain information required for their processing.

It is important to note that *Working Memory* is not the only way by which modules interact as Emotional Mechanisms provide an implicit global communication scheme, as explained in Chapter 4.

Finally, our Architecture also comprises another memory module, which we call Execution Memory. The Execution Memory is responsible for keeping track of the Agent's execution state, allowing the Agent to reason about its past actions and helping it to decide current and future actions.

The Execution Memory stores the sequence of actions performed by the Agents as well as specific events concerning those actions. For example, in the fire-fighting scenario, if the Agent needs to run away from a specific location due to dangerous temperatures, the Execution Memory will already have stored the Agent's last action (e.g.: fighting fire in location X Y) and other relevant information (e.g.: temperature too high, fire propagation high, vegetation density medium). This information will help the Agent decide whether to return to the previous fire fighting location or to choose another more convenient fire-front.

6.3. Goals and the *Goal Manager*

In our Architecture, each Goal is essentially an internal program that consumes proprio-perception information, *Beliefs* stored in *Working Memory* and values of *Emotional State* to produce high-level commands or generate new Goals.

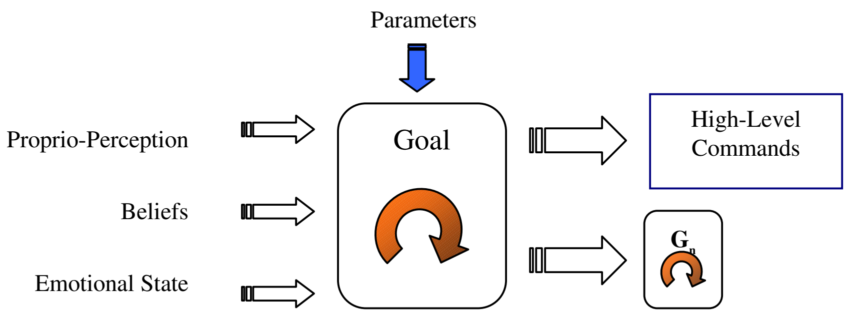

Figure 6.3. Overview of a Goal

Naturally, in such a complex environment, Goals have configuring parameters associated to the control of internal processes. By changing these parameters, it is possible to instantiate multiple versions of the same Goal, with different demands on the quality of the final result and on the computational resources required. For example, when the Agent generates a Goal to approach a given fire-front, there are multiple strategies available for the achievement of such Goal. The Agent may approach the fire-front moving towards the *closest* fire cell or, alternatively, aiming to reach the *safest* nearby position, which will possibly take much longer. Therefore, the Goal provides two options: quicker or safer results. According to the current specific situation (e.g.: energy level, fire size, number of fellow firemen in the area) one of these options will probably be more appropriate.

Each of the various Agent Goals is given processor time by the *Goal Manager* module. The *Goal Manager* maintains a priority queue and allocates processor time to the top priority Goal, which is allowed to run *one* internal cycle. Goals are not pre-empted once processor time has been granted, they are only allowed to run one processing cycle each time. This ensures that goals may run in a pseudo-parallel fashion. However, processor time may be allocated almost exclusively to high priority goals, because their priority does not decrease significantly after being given processor time. In practice, more urgent Goals will run much more frequently.

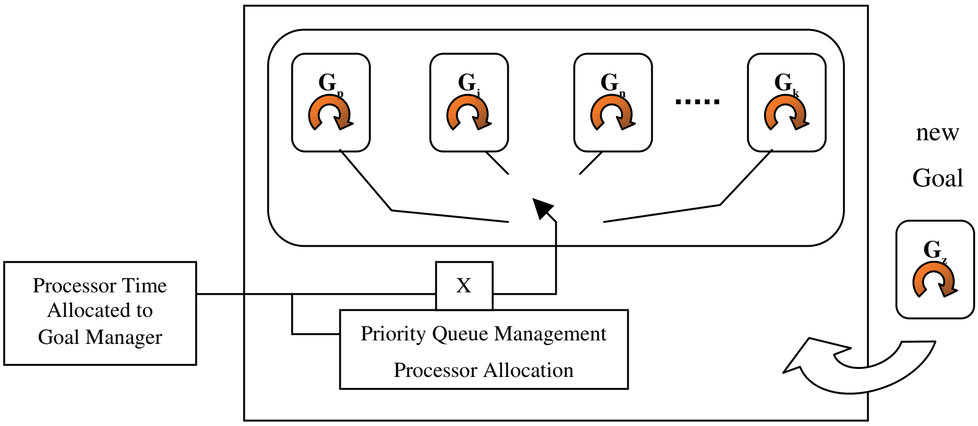

Figure 6.4. The Goal Manager

Goals may request the *Goal Manager* to deactivate other Goals that are incompatible with them. For instance, the Agent will not be able to simultaneously maintain a Goal to fight a fire spot and a Goal to run away. Of course, deactivation is carried out on the basis of priority: Goals may only deactivate other goals if they have a lower priority. Goals may be reactivated after the incompatibility condition disappears. For example, the Goal to run away has obviously higher priority than the Goal to fight the fire and will, therefore, be able to deactivate the later.

6.4. Layered Architecture

Functional Modules are distributed between two different layers. Their exact dispositions may be considered quite loose once modules interact with each other and often cross layer conceptual boundaries. Each layer is responsible for providing processor time to the enclosed *Functional Modules* (*Perceptors*, *Reasoners*, *Managers* and *Controllers*) and deciding how frequently they should be scheduled. The scheduling of each module is done sequentially with a frequency that may change over time, according to the importance of the Goals they help to achieve. However, some Functional Modules are always scheduled to run as they are essential for the Agent's fundamental goal, survival. For instance, the *Goal Manager* and *Perceptors* related with the Survival Goal will always be scheduled to run (although not every Goal will then be scheduled to run by the *Goal Manager*). The scheduling cycle of each layer is called *Execution Cycle*.

Although we have only implemented two layers, it is possible to expand the Architecture to include more layers. In our Architecture, layers should be seen as an aggregation of information processing modules whose overall processing time is tightly connected with a specific set of Goals. Lower-level layers correspond to groups of modules connected to more basic and essential goals. On the other hand, higher-level layers include modules related to less prioritary goals or that depend on information produced in lower-level layers. However, this does not imply by any means that lower-level modules should necessarily be simpler. Nor does it imply that they will probably be implemented by reactive or hard-wired mechanisms, while higher-level modules will all make use of deliberative processes. Of course, this may eventually be the case, but if it happens to be so, it is a result of the requirements imposed by certain goals (e.g.: response time) and not because of an a priori criteria imposed by the designer.

The significant feature of the proposed Architecture is that the management of processing time is done upwards, from lower-level to higher-level layers. Modules related to higher-level layers will only run after those located in lower-level layers, due to the relative importance of goals at each layer. In the best possible case, higher-level modules will run as frequently as lower-level ones, but, usually, they will be given processor resources only a fraction of the number of times that lower-level modules will. In fact, we may consider that a higher-level layer is simply an additional "Module" of a lower-level layer that is chained to run along with the other Functional Modules. The scheduling frequency of such a "Module" should depend on how well the active Goals of the lower-level layer (the most basic ones) are being ensured.

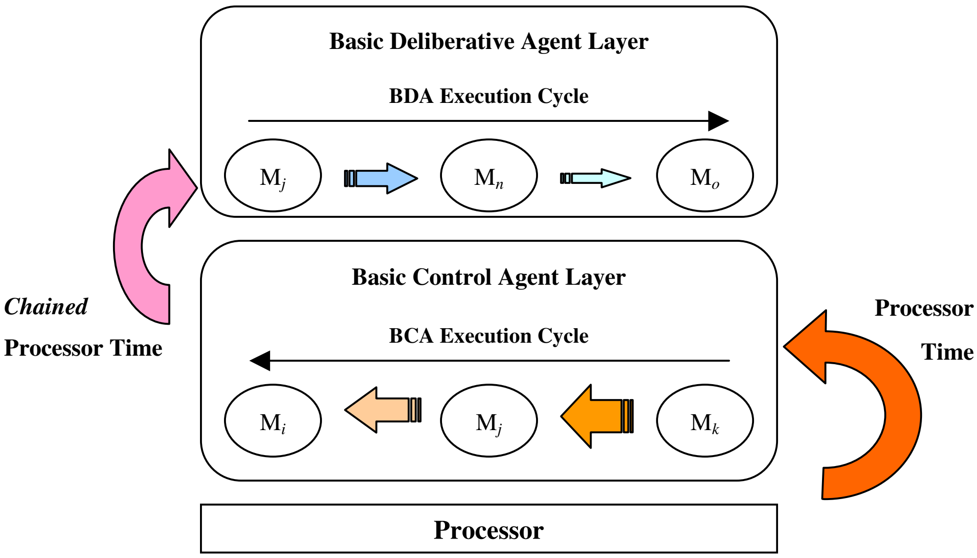

Figure 6.5. Allocation of processor time among Functional Modules in the two layers.

The general idea of this Architecture, thought for environments where Agents have to deal with multiple and simultaneous concerns, is that all modules run in a pseudo-parallel fashion, sharing the existing computational resources according to the current importance of the Goal(s) that they help to achieve. The role Emotions play in this process will become clearer throughout the next sections.

The Basic Control Agent Layer

As we have mentioned before, in our current implementation of the Architecture we have developed two layers. On top of the *Agent Skeleton* layer, we have placed a layer named Basic Control Agent (BCA) Layer which was built around one major Goal: to ensure that the Agent will "survive" in its environment. The notion of "survival" depends, of course, on the environment but, in our very anthropomorphic scenario, it simply means that the Agent must ensure its safety and not get caught by the fire. In order to do so effectively, several *Functional Modules* need to be placed at this layer.

Starting with *Perceptors*, the BCA Layer includes the following specialized *Perceptor* modules:

- *Pain Perceptor* – tracks Agent "energy" to detect rapid decreases that result from harmful events.
- *Temperature Perceptor* – analyzes temperature and its variations.
- *Fire Perceptors* – analyzes close range and medium range visual maps to extract data about the surrounding fire (intensity of burning cells; distances; starting of new fire spots).

The processing tasks associated with these *Perceptors* are executed during each *BCA Execution Cycle*, producing (or revising) *Beliefs* that become globally available through the *Working Memory*. Also located in this layer is the *Goal Manager* module that is responsible for scheduling *all* of the Agent's Goals.

The main Goal at this layer is the Survival Goal. It is constantly monitoring sensory data (temperature, pain), *Beliefs* (intensity and spreading speed of nearby fire) and the value of *Emotional State* (the value of "Fear" Accumulator), i.e. parameters that are relevant or connected with the Agent's survival. Whenever there is a situation that may be very dangerous for the Agent, i.e. to its survival, the Survival Goal will generate another Goal that will make the Agent run to a safer position. These situations are identified by simple (and fast) comparisons made with certain thresholds, such as the maximum sustainable temperature, the maximum safe temperature increase, the maximum safe fire spreading speed or the maximum level of Fear the Agent is willing to reach. Whenever these thresholds are surpassed, the Survival Goal will simply generate a new goal in order to escape to a safer position. The new Goal will be added to the *Goal Manager* queue and run in parallel with the already existing ones.

It is important to emphasize that the Survival Goal has several configurable parameters. Each of the thresholds used to decide whether a new goal should be generated can be changed, promoting different behaviors towards the environment. The Survival Goal is relaxed if thresholds (e.g.: maximum sustainable temperature) are expanded. This will promote risky attitudes from the Agent, as for example, keeping a shorter distance to fire, which may also be important in certain situations. For this particular case, keeping shorter distances to fire-fronts will enable the Agent to fight the fire more effectively. However, these parameters need to be set appropriately so that the Survival Goal will still be ensured.

Another essential module placed at this layer is the Action Manager. The Action Manager is responsible for pipelining the commands issued by Goals (by means of Controller Modules) to the *Agent Skeleton* Layer. The current implementation of the Action Manager does not test for conflicts among commands, such as for example two simultaneous commands coming from different Goals, one requesting the Agent to move backwards while the other is requesting it to move forwards. This will be the subject of future work. For now, there has been great care in avoiding conflicts using the *Goal Manager* to ensure proper coordination between Goals.

Finally, the BCA Layer is also responsible for providing processing time to the upper layer. The *Execution Cycle* of the upper layer, the Basic Deliberative Agent Layer, is chained in the *Execution Cycle* of this layer as if it was another functional module. Therefore, the Basic Deliberative Agent Layer will only be given a share of BCA Layer processing time, which it will have to divide among its own *Functional Modules*.

The following image illustrates the Execution Cycle at the BCA layer:

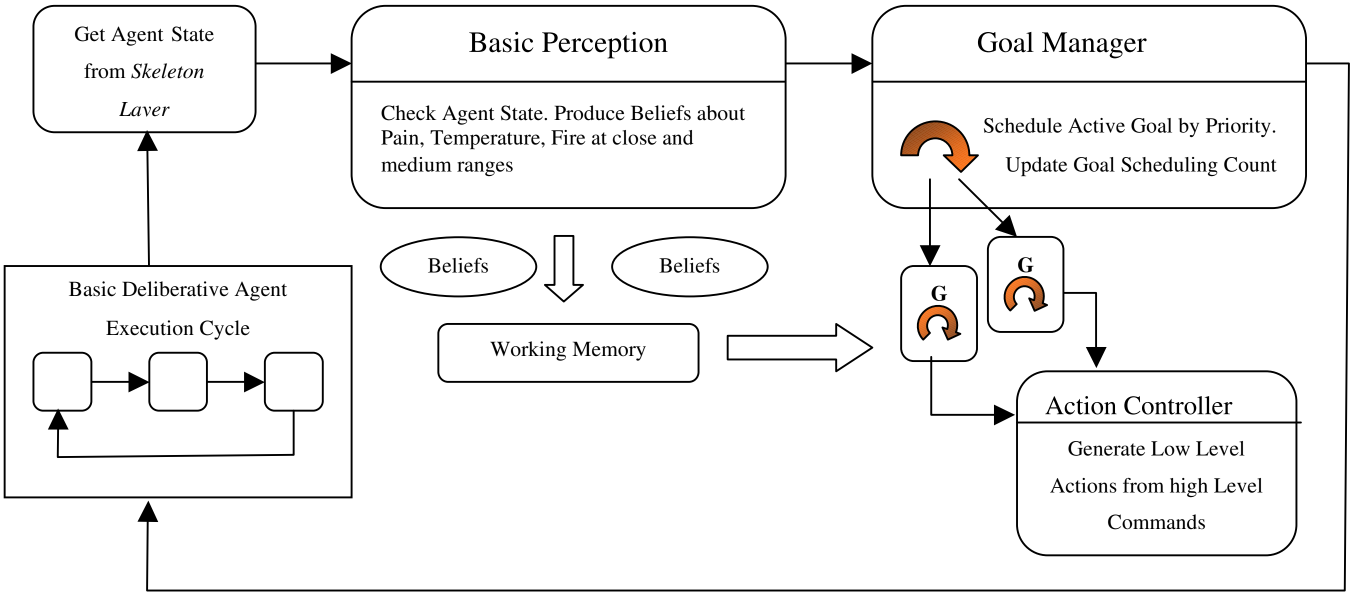

Figure 6.6. Execution Cycle at the BCA layer.

The Basic Deliberative Agent Layer

While the BCA layer is mainly concerned with the most essential Goal of the Agent, survival, the layer placed immediately above, the Basic Deliberative Agent (BDA) Layer, is essentially focused on specific task related Goals. This layer's *Functional Modules* are aimed at providing the Agent with the mental capabilities needed to achieve the Goals related to fire fighting. Of course, these Goals are only reasonable if the Agent fulfils the basic Goal of surviving. Consequently, they should be given processing time depending on the Agent's success in achieving its primary Goal. This may be achieved in two ways:

1. *directly*, by ensuring that the priority assigned to these goals is never higher than the one assigned to Goals in the BCA layer. This way, the Goal Manager will never allocate more processor time to Goals at this layer than it does to Goals at the BCA layer.

2. *indirectly*, by chaining the execution of the BDA Layer in the *Execution Cycle* of the BCA Layer. Thus, additional processing (e.g.: using Reasoners) required for Goal achievement at this layer will only be performed after the processing is executed in the BCA layer, and using only the remaining processor time.

Goals in this layer may be divided in two different groups. In the first group we have included those Goals that are more concerned with direct fire-fighting operations, such as for example goals like "Move to a specified location", "(Re)Approach a given fire segment", "Fight a fire in a close range location". These Goals usually follow naturally from each other and compose the basic fire-fighting sequence. The following picture illustrates these relationships.

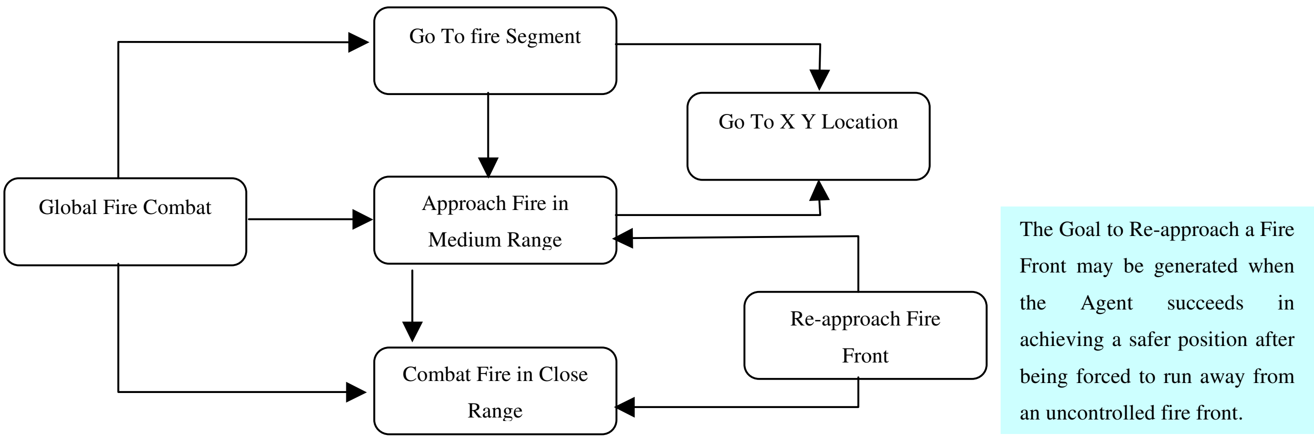

Figure 6.7. Goal Generation flow at BDA Layer.

Goals in the second group are connected with global monitoring of the situation and the corresponding decision-making. In most cases, these goals are responsible for producing the information required by operational Goals. In this second group, we include goals that are constantly being sought, such as "Track fire evolution in close range locations", "Track Fire Evolution in Medium Range Location", "Check if repositioning is needed" or "analyze global fire estimates". Despite their almost permanent nature, these goals' priority is not very high and their scheduling is highly flexible. In fact, the Goal Manager does not need to schedule them in every Execution Cycle because they are concerned with global or distant realities that do not change significantly in short intervals. However, they are important for strategic decisions such as changing current combat position or ensuring that there is always a runaway corridor available. This kind of information is very important for leadership decisions, if we consider a multi-agent approach. The priority of these Goals is configurable according to the role the Agent may have in a fire fighting team.

In order to account for all these goals, the BDA Layer is equipped with *Reasoner* modules capable of processing more complex information and producing higher-level *Beliefs*. These *Reasoners* operate on the data produced by lower-level modules and on information distributed by the Pyrosim simulator, such as a terrain map (pocket map) or a periodic estimate of the fire's global evolution. They also share

Beliefs with each other creating a horizontal flow of information. The BDA Layer includes the following *Reasoners*:

- *Pocket Map Reasoner:* this module is capable of generating path plans between two points in the map, taking into account factors such as the geometry of the terrain, distance to fire, visual occlusion, etc.
- *Fire Evolution Reasoners:* analyze close range and medium range visual maps to extract data about the evolution of the surrounding fire (speed of spread, most dangerous or safest areas, etc.).
- *Fire Map Reasoner*: this module analyzes the global fire estimate in order to detect fire segments (continuous areas), and extract several features about them (size, average intensity, distance, etc.).
- *Fire Segment Evolution Reasoner*: tracks the evolution of fire segments, calculating how fast and where to they are evolving.

Finally, the BDA Layer also includes the Execution Memory module to allow reasoning about previous actions. The information stored in Execution Memory is used to decide new goals coherently (e.g.: re-approach a location the Agent had to run from) and it is important for the Agent in performing evaluation of its own success (e.g.: How many time did I need to run away up to this moment?).

6.5. Emotional State and Emotional Mechanisms

The Agent's Emotional State includes three Emotional Mechanisms, which we labelled using the name of Human Emotional Phenomena, for a better understanding of their functional role:

1. **"Fear"**: Specific Emotion that will be elicited whenever a dangerous or uncontrollable condition (temperature too high or fire too close) is detected. "Fear" is tightly connected to the Survival Goal;
2. **"Anxiety"**: Specific Emotion (or Mood) should be elicited when the Agent faces a possibly difficult situation (e.g.: running out of water, fire-front approaching an important resource).
3. **"Self-Confidence"**: Mood elicited as a result of a successful performance. (e.g.: has successfully extinguished the last fire cells). This Emotion promotes optimistic behaviors.

For each of these Emotional Mechanisms there is an Emotional Accumulator that functions exactly as described in Chapter 4. The values of Emotional Accumulators are globally accessible for inspection throughout the Architecture. Each layer and their corresponding Functional Modules have constant access to the global Emotional State. Before beginning an Execution Cycle, the values stored in the Emotional Accumulators may be used to configure parameters related to the Execution Cycle itself, the operating mode of the associated Functional Modules, and several parameters of corresponding Goals.

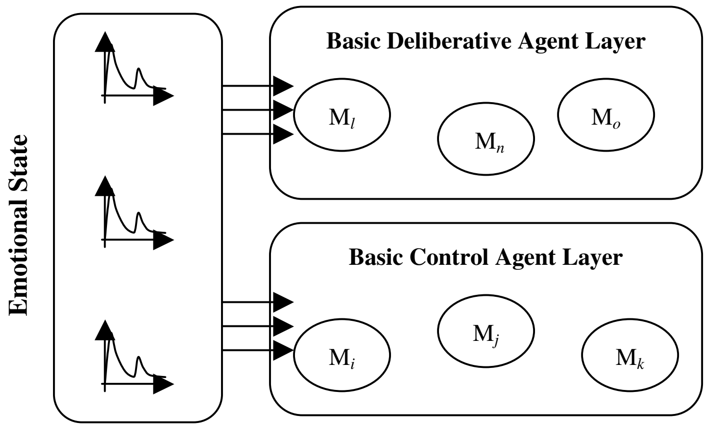

Figure 6.8. Values stored in Emotional Accumulators may be used for several purposes in Layers and Functional Modules.

Emotional Accumulators are updated by Emotional Evaluation functions dispersed all over the Architecture. These Emotional Evaluation Functions are located in both layers and inside Goals, Perceptors, Reasoners and Managers, each contributing to the global Emotional State, according to their scope of influence. For example, the elicitation of "Fear" is deeply connected with local perception. The Functional Module responsible for the perception of nearby fire, the Fire Perceptor, is able to inject impulses in the "Fear" Emotional Accumulator. Therefore, this module contains one of the Emotional Elicitation Functions related to the Fear Emotional Mechanism. However, this is not the only Emotional Evaluation Function connected to "Fear", since it is possible to have others all over the Architecture, for instance, in the Temperature Perceptor or directly in the Survival Goal. The following picture illustrates the various sources of Emotional Evaluation that provide stimuli over a single Emotional Accumulator.

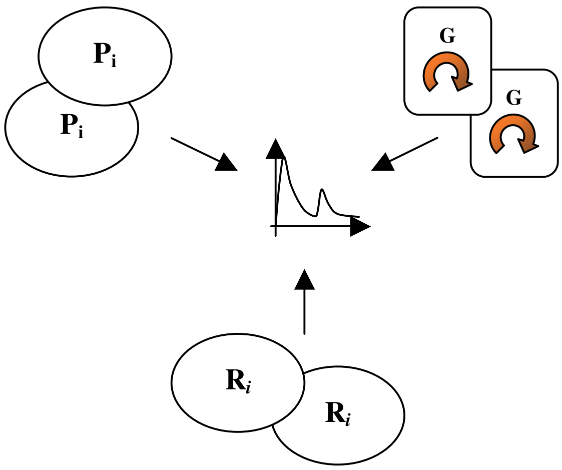

Figure 6.9. Perceptors, P_i, Reasoners, R_i, and Goals may contain Emotional Evaluation Functions that interact with a single Emotional Accumulator

6.6. Architecture Overview

The next figure gives a final overview of the Architecture. The boxes represent generic *Functional Modules* of a given category. The two shaded regions correspond to the BCA and the BDA layers.

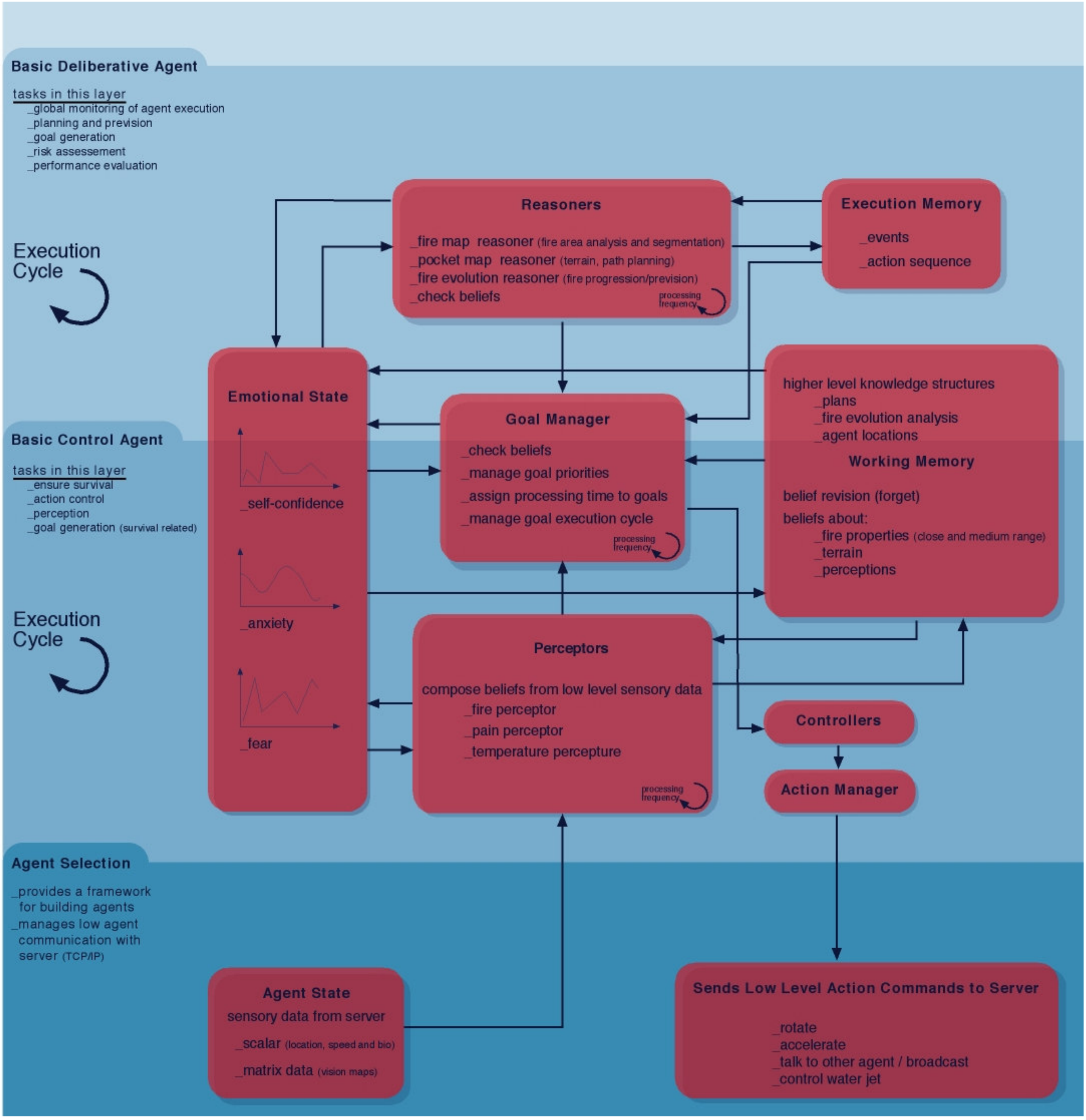

Figure 6.10. Overview of the Emotion-Based Agent Architecture

6.7. Summary and Conclusions

In this Chapter, we have introduced our Emotion-based Architecture focusing mainly on its components. We have also presented all the Functional modules and explained how they are grouped in two layers, the Basic Control Agent Layer and the Basic Deliberative Agent Layer, according to the Goals they help to achieve.

The Basic Control Agent Layer is mainly concerned with basic Goals, such as ensuring Agent survival in the environment. The Basic Deliberative Layer, which is built on top of the previous one, is related to higher-level Goals, and related to fire combat. Each Layer includes several Functional Modules and is responsible for managing its own processor time.

However, in this highly distributed Architecture, processor time has not only to be divided between Functional Modules that run inside each layer, but it also has to be shared by the layers themselves. The Basic Deliberative Layer will be set to run chained on the Execution Cycle of the Basic Control Layer, ensuring that the lower layer basic Goals will always have direct (and indirect) priority in accessing processor time. Additionally, the Architecture holds several configurable parameters. Functional Modules, Goals and both layers have parameters that may be changed.

We have also briefly explained how the Emotional Mechanisms, described in Chapter 4, are integrated within the Architecture. We introduced the three Emotional Mechanisms that are used in the Architecture: “Fear”, “Self-Confidence” and “Anxiety”. We have described how Emotional Evaluation Functions are distributed between all components and how they stimulate individual Emotional Accumulators simultaneously. However, we did not address the details of the interactions established between Emotional Mechanisms and Functional Modules, Goals and Layers. This will be the subject of the next Chapter.

Finally, we have presented a global perspective of the Architecture in which the interaction between all Functional Modules and the layer may be identified more easily.

CHAPTER 7. THE EMOTIONAL MECHANISMS

7.1. Introduction

In this Chapter we will describe the Emotional Mechanisms included in our Architecture.

We will start by revising and emphasizing some of the Architecture's properties that enable the inclusion of Emotional Mechanisms. In particular, we will focus on various possibilities of configuring the Architecture in order to shape different and adapted behaviors, such as those resulting from different Emotional States. We will consider the configuration possibilities at three different levels: Goal, Functional Modules and also Layers.

We will then explain the three Emotional Mechanisms that operate inside the Architecture: "Fear", "Anxiety" and "Self-Confidence". For each of these Emotional Mechanisms, we will address issues regarding Emotional Elicitation, which is made through various Emotional Evaluation Functions, distributed by the several components of the Architecture. We will then address the interaction of Emotional Mechanisms in the configuration of the various parameter types available in the Architectures. Specifically, for each Emotional Mechanism, we will show how the values stored in the corresponding Emotional Accumulator may be used to dynamically adapt the Agent's behavior. We will also relate these interactions with the information processing Strategies, described in Chapter 2. Finally, we will summarize the major points about Emotional Mechanisms presented in this Chapter.

7.2. A Configurable Architecture

Before starting to explain the precise functionality of Emotional Mechanisms, it is important to emphasize the configurability properties of our Architecture at three different (but interconnected) levels: Functional Modules, Goals and Architectural Layers.

The operation of individual Functional Modules is usually highly configurable and dependent on several parameters. For example, the Fire Map Reasoner calculates the path plan using an "oriented" depth-first search in the action-state space. In addition to input data such as terrain maps, and the starting and ending points of the path, this planning procedure involves at least three control parameters:

(i) depth of search, i.e. how many steps should be considered;
(ii) breadth of search, i.e. the number of possible actions (directions) to consider at each step of the plan;
(iii) cost function used to evaluate each step of the plan, i.e. how to account for risk, completion time, physical effort, etc.

Changing the parameters' values has impact both on the final plan (e.g.: quicker plan but involving some risky steps) as well as on computational cost of the search procedure itself (processing time, memory required). For example, increasing the depth of the search will allow the Agent to consider smaller steps and obtain a more detailed plan, but it will also increase its computational weight. On the other hand, increasing the breadth of the planning procedure will allow the Agent to take into account more options at each step, resulting into more flexible plans, but it will also increase computational effort. Finally, altering the cost function will change the quality parameters of the resulting plan. If we increase the weight associated to risk, the plan obtained will be very conservative regarding safety, although it may require the Agent to follow a longer path.

Goals also have some configurable parameters. For example, the Goal "Fight a fire in a close range location" involves three threshold temperatures, T_{MIN}, T_{OPT} and T_{MAX}, which the Agent uses to decide whether it should change its positioning towards the fire or run away to a less dangerous location.

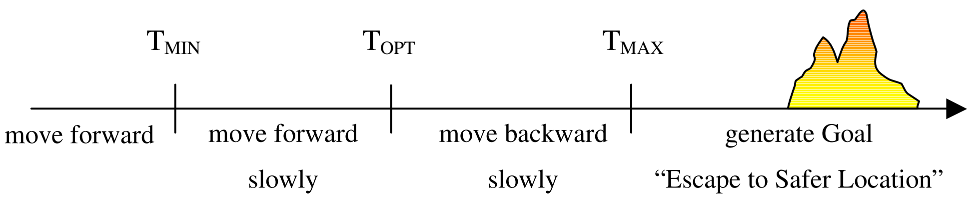

Figure 7.1 Temperature thresholds and decision-making. Threshold temperatures are configurable.

Varying these values will change the average distance the Agent keeps to the fire. This has direct consequences on the global efficiency of fire combat, since the efficiency of the water jet decreases with distance from target. Fighting the fire from a closer position is advantageous but it represents a significant risk for the Agent.

However, this is not the only goal that may be configured. The next table shows some of the parameters associated with this and other Goals.

Table 7.1 Summary of parameters available for some of the Agent's Goals

Goal	Parameters
"Combat a fire in a close range location"	3 threshold temperatures, maximum allowable distance to fellow firemen
"Escape To Safer Location"	Minimum Distance from fire, Safety Temperature
"(Re)Approach Fire Front in Medium Range location"	Safety level and distance to fire of target position
Combat Fire (globally)"	Area and spreading speed of fire segment to be approached, maximum growth before choosing an alternate fire location.

There is also the possibility of varying the internal behavior of each layer. Layers are responsible for deciding which functional modules will be scheduled to run in each execution cycle. In practice, layers are able to decide how much time is spent in analyzing perceptions, in controlling current actions, in planning future actions, in general decision-making or in *exploring* possible scenarios. For instance, Perceptors concerned with the Agent's close surrounding may be scheduled to run on every Execution Cycle, whereas those concerned with events happening further away may be scheduled to run only once in every ten cycles (especially if the current computational effort of the Agent is increasing). More generally, since the Execution Cycle of the Basic Deliberative Agent Layer is directly chained to the Execution Cycle of the underlying Basic Control Agent Layer, the total computational effort spent on "higher-level" tasks or Goals may be controlled by setting an appropriate scheduling cycle for the whole BDA Layer.

The next picture illustrates how processor time is shared between the tasks (performed by Functional Modules) in each layer.

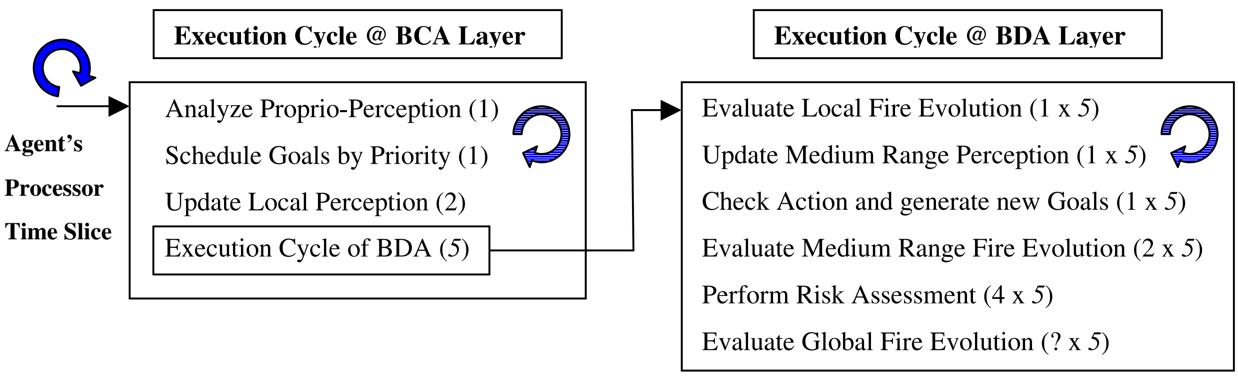

Figure 7.2 An example of how processor time can be allocated between Layers and inside each Layer.

The numbers in brackets indicate the scheduling period of each Functional Module. In this example, the analysis of Proprio-Perception will be executed in every Execution Cycle, whereas the updating of local perception will be scheduled to run only in half the Execution Cycles. Also, the Execution Cycle of the BDA will be set to run only once for each five BCA Execution Cycles. Therefore, tasks related with risk assessment would be scheduled to run only one twentieth of the times than tasks related to analyzing Proprio-perception. However, all individual scheduling parameters may be changed enabling a complete alteration of the Agent's global scheduling policy.

There are many possible configurations for the parameters available at three levels (Functional Modules, Goals and Architectural Layers). It is necessary to have an efficient method to set those parameters coherently over the entire Architecture. It would be particularly interesting to obtain configurations that efficiently allocate the Agent's limited processing capabilities according to the demands of the environment: processing resources should be spent where most needed and Functional Modules and Goals should be dynamically adapted to circumstances. In our Architecture, such configuration is performed with the help of Emotional Mechanisms.

We have included in our Architecture three different Emotional-Mechanisms, which, for the purpose of visualization, we will be referring to by the functionally corresponding Human Emotions: "Fear", "Anxiety", and "Self-Confidence".

7.3. Fear

7.3.1. The Elicitation

We start by analyzing the "Fear" Emotional Mechanism because it is probably the most immediately applicable to the Architecture. "Fear" is elicited whenever a possibly dangerous or uncontrollable situation is detected, i.e. whenever the Survival Goal is at stake. "Fear" should reflect the Agent's inability to cope with the current situation with its current resources. For instance, if the temperature becomes too high or if it increases suddenly, the "Fear" level should be increased because the Agent is probably facing a survival-threatening situation. There are multiple other conditions that indicate that the Agent is, or may be facing in the near future, a very dangerous situation. The next table lists some of these situations.

Table 7.2 Summary of eliciting conditions for "Fear"

Nature	Condition
Temperature	High temperature or rapid increase in the temperature
Fire	Fire in the same cell. Distance to fire. Fire intensity in close and medium range locations are rapidly increasing.
Pain	High Level of pain
Help Availability	No fellow fireman nearby when combating a fire-front
Energy	Low levels of Energy

All theses conditions are detected by different specialized Perceptors. Therefore, the "Fear" Emotional Accumulator is updated by several Emotional Evaluation Functions that exist inside such Perceptors. The next figure illustrates various sources that contribute to Fear Emotional Accumulator:

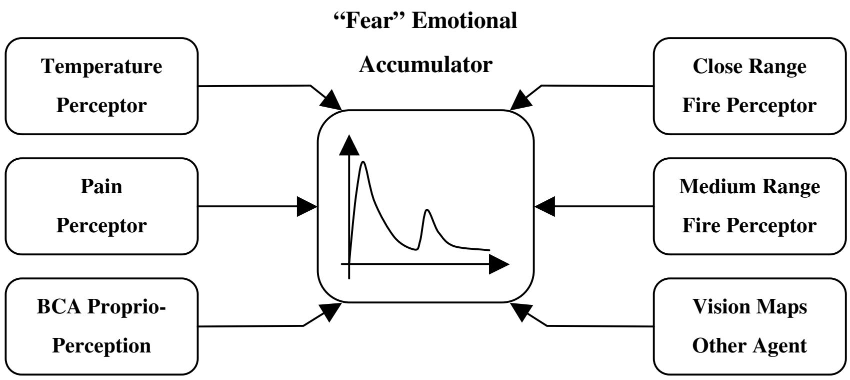

Figure 7.3 Elicitation source for the "Fear" Emotional Mechanism

A fragment of code similar to the following snippet could be found inside the Close Range Fire Perceptor to implement its Emotional Evaluation Function:

```
if  (agent_terrain_cell_is_burning) increase_Emotional_Accumulator(Fear,2);
if (fire_intensity_increase > safeIncrease) increase_Emotional_Accumulator(Fear,0.5 * fire_intensity_increase);
if (number_of_igninted_cells > 0) increase_Emotional_Accumulator(Fear,1);
```

7.3.2. The Interactions

In dangerous situations, it seems reasonable that the Agent should concentrate its resources in coping with the possible threat. The processing time assigned to analyzing local data, which is where the threat is most certainly originated from, and to controlling movement should be increased. It is also important to increase the frequency of perception updates to ensure that the Agent is operating using the most recent data and is not missing any important detail. On the other hand, less urgent Goals (e.g.: global monitoring of fire evolution) and certain higher-level modules should be scheduled less frequently to decrease competition for processor time. This could be done simply by instructing the Goal manager only to schedule goals whose priority is higher than a certain minimum priority threshold.

Assuming that the "Fear" Emotional Mechanism is actually identifying possible dangerous situations, even if sometimes it is too conservative and incorrectly identifies as dangerous a situation that is not dangerous, we may use the value it stores to control architectural parameters to achieve the effects explained before. The following snippet of pseudo-code illustrates how this could be implemented directly from the BCA Layer:

```
IF (FEAR.level >= MEDIUM) THEN
{
  CLOSE_RANGE_PERCEPTOR.frequency++;
  BDALayer.scheduling_frequency--;
  GOALMANAGER. min_scheduling_priority++;
```

}

The “Fear” Emotional mechanism also has effects at other more specific levels. For example, parameters related to path planning are changed as a function “Fear”. In the Pocket Map Reasoner, whenever “Fear” is high, the complexity of the planning procedure is reduced in order to obtain a plan more rapidly: both depth and breadth of the associated search algorithm are decreased. The plan thus obtained is possibly not the “best” one, but it is certainly good enough given the urgency of the situation and the probably scarce processing resources available at that moment.

In addition, since “Fear” signals that the Agent is facing a dangerous situation, it seems natural that it should adopt a more pessimistic posture towards the environment for a certain period. Acting pessimistic will make the Agent avoid other risky situations. For instance, the distance the Agent will keep to the fire depends on the surrounding temperature. Initially, the Agent is set not to run away from fire until the temperature reaches 65º C (T_{MAX}), allowing him to fight fire from shorter distances and, therefore, more efficiently. However, if “Fear” is high, which may result from a previous fearful episode, the threshold temperature is decreased and the Agent will start moving back sooner, keeping a larger and safer distance to fire. The Agent is thus being pessimistic regarding its own capabilities to fight fire, but this will certainly reduce the risk of being caught by it. Note that the “Fear” Emotional Accumulator will decay with time and if no more fearful episodes occur, the pessimistic attitude will tend to disappear.

The next table summarizes some of the interactions that Fear is able to produce at three different levels:

Table 7.3 The effects of “Fear”

Level	Object	Changes	Effect
Goals	“Combat Close Range Fire”	Reduce threshold temperatures (T_{OPT} and T_{MAX}) Reduce distance to fellow firemen	Keep a safer positioning
	“Escape To Safer Position”	Increase minimum distance from fire-front that is considered safe.	Ensure a reasonable distance from the fire-front to allow a better re-approach
	“(Re-)Approach Medium Range Fire”	Choose a safer target position (e.g.: favorable terrain inclination)	Avoid another risky situation
Modules	Pocket Map Reasoner (Path Planner)	Reduce depth and breadth of planning procedure.	Reduce complexity of the planning Procedure; Obtain simpler plans faster.
Layers	BCA	Increase the Frequency of (local) Perceptors; Increase the Minimum priority needed for Goals to be scheduled; Decrease the Scheduling Frequency of BDA.	Concentrate resources at the level of BCA; update local perception as fast as possible; focus on Basic Goals
	BDA	Increase the Priority of the Goal “Track Fire Evolution in Visible Range”	Keep the estimate of fire evolution reasonably updated

7.3.3. Additional remarks

Fear is a negative valence Emotion. It signals a strong inability to cope with the environment. In Chapter 2, we identified four different information processing strategies and related two of them to negative valence Emotions, namely *Motivated Processing* and *Substantive Processing*. In the case of "Fear", where a specific Goal is at stake, survival, one would expect that a Motivated Processing Strategy should be employed. As mentioned in Chapter 2, *Motivated Processing* is employed when the Agent has a specific set of Goals that motivate all processing and usually requires a great amount of computational effort.

In fact, if we consider what has been described in the previous sections, we may verify that the Agent's global behavior under the influence of "Fear" Emotional Mechanism is compatible with the definition of *Motivated Processing*. The Agent focuses all its processing resources around one specific Goal, Survival. At the same time, there is a shift in the allocation of processor time to analyzing data about the Agent's surroundings and fire evolution in close and medium range.

Additionally, when under the influence of negative Emotion, Agents tend to adopt a conservative or even pessimistic attitude towards the environment. In the case of "Fear", we may easily identify such an attitude when there is a reduction of the temperature thresholds, which results in a more distant positioning to fire. Finally, Substantive Processing does not make significant use of any Emotional Information. Again, our Agent will try to use only information it extracts from the environment, which it tries to keep as updated as possible. In practice, the information about close and medium range surroundings is almost all the information it needs to ensure the Goal of Survival.

7.4. Anxiety

7.4.1. The Elicitation

"Anxiety" is an Emotional Mechanism related to the anticipation of situations that may become difficult to cope with. Contrary to the case of "Fear" that is connected to *very urgent* or immediate life-threatening situations, "Anxiety" results from the evaluation of the environment and of the Agents' own capabilities concerning future possibly threatening situations. For example, "Anxiety" will increase whenever the Agent receives information about the existence of large fires in a nearby location (environmental factor) because a difficult situation will possibly be around. If the Agent starts running out of water (Agent internal factor), Anxiety will also increase. Other factors include the energy level, wind speed, terrain geometry, number of surrounding fellow Agents and Beliefs produced by Reasoners, particularly those related with Fire analysis (e.g.: Fire Map Reasoner).

Table 7.4 Summary of eliciting conditions for "Anxiety"

Nature	Condition
Wind	High wind speed. Irregular direction.
Geography and Agent Location	Irregular Geography. Being located in a point above the level of a nearby fire.
Fire Analysis and Reasoning	Large areas burning (even far away). Fire spreading very quickly in medium range locations. Fire approaching strategic locations
Help Availability	Small team for the size of the fire
Proprio-Perception	Low levels of Energy. Running out of water
"Fear" Emotion	Successive Fearful episodes

As shown in the previous table, "Anxiety", as "Fear", has multiple elicitation sources. However, "Anxiety" is connected to more indirect conditions that do not pose immediate threats. It may thus be considered an Emotional Mechanism closer to Capabilities and Goals, located in the Basic Deliberative Layer, contrary to the Fear Emotional Mechanism that is closer to the Basic Control Agent Layer.

It is interesting to note that "Anxiety" is also influenced by the "Fear" Emotional Mechanism. In fact, if the Agent is going through successive fearful episodes, it is very probable that it did not adapt to the current environment and, therefore, it is very natural that it might find another threatening situation in the near future. Successive high levels of fear are therefore conditions that should contribute to increase the "Anxiety" level.

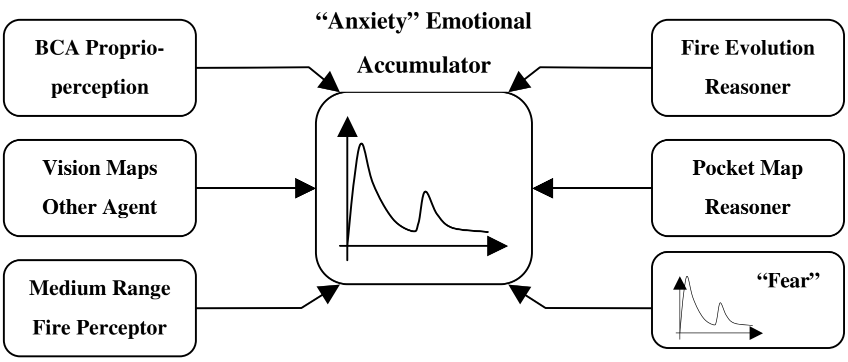

Figure 7.4 Elicitation sources for the "Anxiety" Emotional Mechanism

7.4.2. The Interactions

In a certain way, "Anxiety" works as a "preventive" mechanism. High "Anxiety" levels will promote a very intensive processing state so that the Agent will be able to detect possible Goal threatening situations or favorable opportunities. The Agent will invest a great deal of its processing resources in analyzing perception data and producing higher-level Beliefs to use in the decision of new

Goals. This is generally achieved by increasing the execution frequency of de Basic Deliberative Layer, where Reasoners are scheduled to run.

As a result, the Agent will be configured to produce updated Beliefs about the areas where the fire is progressing faster, which are the most dangerous ones (where fire seems to be converging), and which seem more worth fighting (e.g.: less damaged). The Agent will also increase the breadth and depth of the path planning procedure in order to increase the final plan's quality. At the same time, plans are revised frequently to integrate the knowledge of the updated Beliefs produced by Reasoners.

The increase in the processing load needs to be compensated by reducing the processor time allocated to other Functional Modules. Since the Agent is not going through a very urgent situation, which would be signaled by "Fear", it is possible to decrease the frequency of the Perceptors and Controllers and thus relax the load of the Basic Control Layer. This extra computational power may now be spent in the Basic Deliberative Layer and be allocated to Reasoners. Clearly, this situation may not be sustainable for long periods as the Agent will eventually need to focus all its resources on a nearby fire spot. However, as we have seen previously, "Fear" will take care of that.

Table 7.5 The effects of "Anxiety"

Level	Object	Changes	Effect
Modules	Pocket Map Reasoner (Path Planner)	Increase depth and breadth of planning procedure. Increase the weight of safety parameter in the cost function.	Test a great number of possibilities to obtain the "best" plan
	Fire Evolution Reasoners	Turn on available processing options.	Increase detail in the analysis procedure; Produce estimates about fire progression.
Layers	BCA	Decrease all Perceptors Frequency; Increase BDA Scheduling.	Release resources at the level of BCA, Give processing Time to BDA
	BDA	Increase all Reasoners Scheduling Increase the Priority of the Goal "Track Fire Evolution in Visible Range"	Keep the estimate of fire evolution reasonably updated

There is a certain amount of overlap in the effects of "Anxiety" and "Fear". However, for example, the planning procedure is changed in opposite ways by "Fear" and "Anxiety". The behavior of the BCA layer in the case of "Fear" is also almost the opposite of the one in the case of "Anxiety". Obviously, it will not be possible to change simultaneously all available parameters in opposite ways. However, since the Emotional State is globally available to all elements in the Architecture, some additional rules may be imposed when incompatible cases arise. And because "Fear" is directly connected with urgent situations, it should always subsume "Anxiety" in its effects. The following snippet of code illustrates such condition:

```
IF (ANXIETY==HIGH) AND (FEAR < MEDIUM)
THEN PLAN_BREADTH = HIGH
```

The value stored in the "Fear" Emotional Accumulator will naturally decay with time and, when it drops below a certain level, "Anxiety" will be able to produce its effects.

7.4.3. Additional remarks

"Anxiety", in a certain way, is a negative Emotion as well. Like "Fear", it should promote a certain conservative behavior in the Agent because, although no direct threat is at stake, the situation will probably become difficult to deal with in a close future. Among the four information processing strategies described in Chapter 2, Anxiety would be related to *Substantive Processing*. When performing under *Substantive Processing* information processing strategies, Agents operate in a systematic fashion with complex information processing structures, without having a specific goal to attain at that moment.

As described in the previous sections, when the "Anxiety" level rises, the Agent responds by increasing the amount of processor time allocated to the BDA Layer and, at the same time, increasing the complexity of planning procedures and fire evolution analysis. In our opinion, this clearly corresponds to adopting a *Substantive Processing* strategy.

Additionally, *Substantive Processing* strategies make a significant use of Emotional Information in the overall processing. However, that is not so visible in our Architecture because the value of the "Anxiety" Accumulator itself is not directly used as an input for any process. We may identify a relationship between "Anxiety" and "Fear" in the elicitation process of "Anxiety". This should not be seen as an example of the *Emotion as Information* paradigm. On the other hand, the subsumption rules described at the end of the previous section may be seen as an indirect application of such paradigm.

7.5. Self-Confidence

7.5.1. The Elicitation

"Self-Confidence" is an Emotional Mechanism directly related with the success achieved in previous Goals. If the Agent is successfully accomplishing Goals, the level of Self-Confidence should be increased to reflect that the Agent has enough capabilities to cope with the environment.

There are a few situations that might indicate that the Agent is having success in its main goal, which is fire fighting. Firstly, an Agent (or group of Agents) that is able to diminish the intensity of fire, or even extinguish the fire, at the cell that it is currently fighting, should receive a positive indication about its fitness to cope with the fire. Such an indication should be materialized in an increase in the "Self-Confidence" level. Similarly, if fire at close and medium range locations is receding due to successive fire extinguishments recently achieved, the Agent should also get an increment in its “Self-Confidence” to reflect its capability to cope with the environment. The level of “Self-Confidence” should be changed, even if the Agent is not having direct influence in a positive outcome, such as for example a significant decrease in the intensity of a distant fire segment as a result of a wind change or specific terrain geometry. In this case, the level of “Self-Confidence” should also increase because the environment has globally become more tractable, even if this does not represent an immediate advantage for the Agent.

Additionally, if other external conditions arise, helping or enabling the Agent to be more successful in fire fighting, the level of “Self-Confidence” should also be increased. For example, whenever the concentration of firemen Agents increases in the surroundings of a certain location, the global fire fighting potential in that area grows. Consequently, each of those Agents should feel more comfortable to face the current situation and their level of “Self-Confidence” will increment.

On the other hand, situations such as moving back or running away from a fire-front should have opposite effects on the level of “Self-Confidence” than the ones mentioned before. If, when combating a given fire cell, an Agent constantly needs to move back, this may mean that he is not able to cope with the fire-front. This is especially true if the Agent is forced to run away from its location often. In both cases, the Agent is obviously unable to cope with the situation. Therefore, the level of “Self-Confidence” should reflect this fact and be decreased. Another indirect indication of the possible inability to cope with the environment may be drawn from the value of the “Fear” Emotional Mechanisms. Multiple fearful episodes mean that the Agent is frequently going through very difficult situations and this will happen when the Agent is not capable of dealing with the situations that are constantly arising. Thus, the level of “Self-confidence” should decrease whenever the level of “Fear” reaches a given threshold.

It is interesting to note that the detection of most of these situations requires the ability to track the evolution of the environment. Therefore, “Self-Confidence” is an Emotional Mechanism that is tightly connected with Reasoning Capabilities, i.e., with the Basic Deliberative Agent Layer. We had already suggested this relationship when we presented a global overview of our Architecture, in Chapter 6. The “Self-Confidence” Emotional Mechanism is placed inside the Basic Deliberative Agent Layer. The next picture tries to summarize the factors that contribute to changes in the Emotional Accumulator’s values:

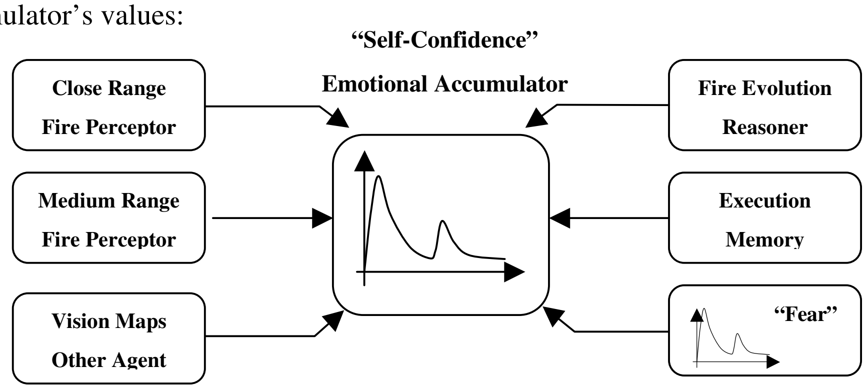

Figure 7.5 - Elicitation sources for the “Self-Confidence” Emotional Mechanism

7.5.2. The Interactions

Therefore, high-levels of “Self-Confidence” signal that the current environment poses no significant difficulties to the Agent. In such a situation, the Agent could simply relax its information processing strategies. The processing resources thereby released may be used in other tasks or Goals that are not urgent but that may become advantageous later. For example, if the Agent is not having problems in extinguishing successive fire cells, some of the processor time that was being spent in analyzing local data could be allocated to analyzing the fire progression in the medium range surroundings. This will help the Agent to find other possible threats or opportunities. Another possible use for the released processor time is to apply it in Learning. This is a promising possibility that we are still experimenting [Mou03].

Moreover, if the Agent is actually having success systematically, then it should adopt a more optimistic approach towards the environment and stretch its own previous limits. In our Architecture, high values of “Self-Confidence” increase the temperatures that the Agent is willing to withstand before starting to move back or run away. As a result, the Agent will remain closer to fire and will be able to fight it much more efficiently. In this way, “Self-Confidence” helps the Agent in having a more “aggressive” behavior in the search for favorable, yet unknown, opportunities. Additionally, Agents will also loose the normal constraints regarding distance to fellow firemen. Usually, firefighters should keep close to each other to protect themselves better and to concentrate their efforts in extinguishing one fire cell quickly. High levels of “Self-Confidence” will increase the maximum allowable distance between firemen resulting in a wider combat front, which is preferable, if possible. This also suggests that high levels of Self-Confidence are useful for experimenting new possibilities. For example, if firemen are feeling comfortable with the current team behavior (signaled by high levels of Self-Confidence), then it seems appropriate to try at that moment a more risky, but possibly more effective, combat tactic. This could lead to quicker fire extinguishments, or to the “discovery” of new effective tactics to control fire. However, during this work, we have focused mainly on the behavior of individual Agents and we did not explore the realm of team coordination.

On the other hand, “Self-Confidence” decreases whenever the Agent is forced to run away. Several unsuccessful attempts to extinguish fire cells mean that the Agent is not capable of coping with the current situation. “Self-Confidence” will be decreased which will have opposite effects to those described before, but will also trigger a *Goal revision process*. A new Goal, more adapted to current Agent possibilities, needs to be generated. For example, the Agent might change the combat strategy or move to a more favorable location. In our current implementation, the Agent will move to a region where the fire is progressing slower. Note that there is a significant cost regarding locomotion as the Agent will not be able to fight fire when moving, causing it to spread faster. Therefore, “Self-Confidence” is an interesting trigger for Goal revision because it includes information about the Agent’s previous successes and not just about the immediate state of the environment.

The following table tries to summarize the effects of high “Self-Confidence” levels.

Table 7.6 The effects of (high) "Self-Confidence"

Level	Object	Changes	Effect
Goals	"Combat Close Range Fire"	Increase temperatures (T_{OPT} and T_{MAX}) Increase distance to fellow firemen	Approach Fire front; widen fire combat front Resist more time before running away.
Modules	Fire Evolution Reasoners	Turn on available processing options.	Increase detail in the analysis procedure; produce estimates about progression of fire
Layers	BCA	Decrease the all Perceptors Frequency of all Perceptors; Increase BDAthe Scheduling Frequency.	Release resources at the level of BCA, Give processing Time to BDA
	BDA	Increase all Reasoners Scheduling or allocate time to Learning procedures[10]	Do not give up current Goal (current fire-front). Explore new possibilities (different team positioning).

7.5.3. Additional remarks

Contrary to the other two cases, "Self-Confidence" is a positive valence Emotion. Therefore, it should promote one of the two minimum-effort processing strategies mentioned in Chapter 2, either *Direct Access* or *Heuristic Processing*. This is a justifiable response, because if the Agent is having success with the environment, it may relax its information processing strategies (e.g.: *Motivated* or *Substantive Processing*) and assign the released resources to other less urgent tasks or Capabilities. The difference between *Direct Access* and *Heuristic Processing* lies in the amount of Emotional Information used and the complexity of the environment where the Agent is operating. Heuristic Processing makes use of a large amount of Emotional Information and is intended for complex environments where the Agent may not be able to employ pre-existing knowledge structures or behaviors. On the other hand, *Direct Access* relies much more on previous experiences that it tries to apply directly without making use of Emotional Information.

In the case of our Agent, the effects of the "Self-Confidence" Emotional Mechanism would place it somewhere closer to *Heuristic Processing*. In this case, we are using the value of the Emotional Accumulator for triggering (not modulating) a Goal revision process, so there is a clear example of the Emotion as Information paradigm. In fact, since the environment is quite complex, it would be very difficult to apply other more *direct* decision techniques, such as rules about the state of the environment, because they would be too complex to obtain and possibly too expensive to check in real-time. It would be very difficult to define the exact conditions that indicate that the Agent should generate a new goal to move to another fire-front or alternatively keep its current positioning.

---

[10] In this work, we have not addressed directly the issue of Learning, although some simple experiments have been made using Pyrosim platform. For a preliminary study, please refer to [Mou03].

Additionally, although we have not implemented it yet, the released processing resources should be used to promote an explorative behavior. Agents under the influence of "Self-Confidence" should be able to try variations of tactics, even if they result in less safe options. Of course, these decisions would have to be taken by a leader in the context of a team effort. We have not explored the issue of team coordination in this work.

7.6. Summary and Conclusions

In this Chapter, we have presented three Emotional Mechanisms included in our Architecture: "Fear", "Anxiety" and "Self-Confidence". We have described the conditions by which such Emotional Mechanisms are activated, namely through various Emotional Evaluation Functions spread all over the Architecture. Each Emotional Evaluation Function is able to detect specific situations that may pose a threat (in the case of "Fear" and "Anxiety") or may suggest an opportunity (in the case of "Self-Confidence"), indicating thereby that the Agent behavior should be readapted to the environment. For each Emotional Mechanism we have demonstrated how such conditions could be detected from data gathered through Perception.

We have then demonstrated how the values stored in Emotional Accumulators, which resulted from the contributions of Emotional Evaluation Functions, could be used to change specific parameters of the Architecture at three different levels, Goal, Functional Modules and Layers, in order to adapt Agent's behavior to the changing environment. Modification of Architecture parameters is achieved in real-time and allows the Agent to dynamically adapt its Resources and Capabilities to the current situation. This adaptation aims at getting the most out of the Agent's Capabilities, and revealing therefore a very functional property of Emotional Mechanisms.

Additionally, for each Emotional Mechanism, we have tried to establish a relationship between the information processing pattern they generate and Human *Information Processing Strategies,* described in Chapter 2. We have identified that "Fear" promotes a processing strategy similar to *Motivated Processing*, and "Anxiety" to *Substantive Processing*. Both of these strategies are extremely resource consuming which is compatible with the need to employ all available resources when difficult situations arise. On the other hand, "Self-Confidence" promotes a processing pattern that is closer to *Heuristic Processing*. In this case, information processing procedures are relaxed and there is a considerable use of Emotional Information in the decisions taken (e.g.: move to another fire-front if "Self-Confidence" drops below a given value).

Finally, the essential point that needs to be emphasized is that Emotional Mechanisms belong to the realm of highly configurable Agent Architectures, which are usually related with very complex environments. The role of Emotional Mechanisms is mainly that of adapting the global information processing capabilities of the Architecture in real-time, as well as its individual Functional Modules.

CHAPTER 8. CONCLUSIONS

8.1. Overview

The work reported in this thesis addresses a complex topic: Emotional Mechanisms. Our goal was to develop a Software Agent Architecture in which Emotional Mechanisms could be used as a functional advantage to Agents, especially for those operating in complex environments.

A great deal of our work has been devoted to exploring a vast amount of dispersed bibliography about Human Emotion so that a deeper understanding of Emotional-Mechanisms and their interaction with Cognition could be achieved. Much of this research about Human Emotion has been summarized in Chapter 2.

At the same time, we investigated other Models and Architectures of Emotional Mechanisms that have been developed in the field of AI. As described in Chapter 3, a significant amount of work has been produced in which Emotional Mechanisms play a functional role. However, most of the work done so far, usually addresses one or only few very specific topics regarding Emotional Mechanisms. To our knowledge a complete approach has still not been developed.

All this research allowed us to identify the building blocks of Emotional Mechanisms and understand how they could interact with other elements that compose generic Software Agent Architectures. Our perspective has always been that all the modeling effort should have a functional purpose in mind. Using this approach, we have developed a general model of Emotional Mechanisms and of their functional interactions, introduced in Chapter 4. We believe that this general model is one of the most important results of our work.

In order to test our models, we decided to develop a software platform for simulation upon which we would develop the Emotional Agent Architecture. The platform developed enables the simulation of a fire-fighting environment, which is complex enough to justify the employment of Emotional Mechanisms. This platform also offers a software layer (the Agent Skeleton Layer) that provides all the basic functionalities needed to build specialized Agents. We described the platform and the simulated environment in Chapter 5.

The simulator allowed us to develop a concrete Emotional-based Agent Architecture with a specific scenario in mind. The Architecture developed is rather general and can be transposed to other application scenarios. It is composed of several Functional Modules that are grouped in two layers, according to the Goals they help to achieve. The Basic Control Agent Layer is built on top of the Agent Skeleton Layer (provided by the simulation platform) and is connected to the Agent's Fundamental Goal in the fire-fighting environment, survival. This layer possesses all the necessary capabilities to address this Goal. At the same time, this layer is responsible for managing and providing processor time to the layer placed immediately above, the Basic Deliberative Layer, which is connected with higher-level goals and Capabilities.

An important conclusion is that this specific layered Architecture is highly configurable and can be set to operate under many possible modes. Configuration may be done at three levels by changing the parameters associated to (i) individual Functional Modules, (ii) Goals and (iii) Architectural Layers. Details about the Architecture may be found in Chapter 6.

The Emotional Architecture includes three different Emotional Mechanisms, namely "Fear", "Anxiety" and "Self-Confidence". Each Emotional Mechanism tries to address specific environment situations, either threats or opportunities, and alters some of the Architecture's parameters available at the three aforementioned levels. These changes are intended to adapt the Agent's current Capabilities to the specific state of the environment. The exact conditions that lead to the elicitation of a given Emotional Mechanism are identified by several Emotional Evaluation Functions, distributed throughout the Architecture's elements. The values stored in the corresponding Emotional Accumulators are used to guide the parameters of the Architecture configuration. Some experiments with simple scenarios lead us to the conclusion that considering Emotional Mechanisms really matters regarding the Agent performance. These Emotional Mechanisms were detailed in Chapter 7.

In summary, the main results of this work are:

- a generic Functional Model of Emotional-Mechanisms;
- an Emotion-based Agent Architecture where Emotional Mechanisms have a functional purpose;
- the establishment of a clear connection between Emotional Mechanisms and specific components of distributed mentalist-like Agent Architectures;
- the application of three Emotional-Mechanisms, whose functionalities are strongly related with Agent's performance, in a specific application scenario: "Fear", "Anxiety" and "Self-Confidence";
- a complete simulation platform based on a relevant real-world problem.

8.2. Current State of Implementation and Limitations

As of this writing, we have implemented most of the Architecture and most of the Emotional Mechanisms. All the Functional Modules described in Chapter 6 were implemented. Additionally, methods for configuring the parameters of Functional Modules, Goals and Layers have been provided.

We have also implemented the mechanisms related with the elicitation of “Fear” and with most of the interactions described in Chapter 7. We have also implemented the most significant features about the “Self-Confidence” Emotional Mechanism, specifically those related with the Goal revision process. However, we have not yet implemented the “Anxiety” Emotional Mechanism, nor have we developed means to clearly visualize Emotional interactions. We are currently trying to solve this last issue.

As for the Pyrosim platform, it is globally functional and all the features described in Chapter 5 have been implemented. In addition, an Agent Programming Manual has been produced and is provided with the Platform to help programming new Agents in Java.

8.3. Future Research Direction

There is still a lot of work to be done in order to refine and sophisticate the proposed Emotion-based Architecture. Besides refining current Emotional Mechanisms and the set of their possible interactions, it would be interesting to study and develop other Emotional Mechanisms such as “Frustration” and “Angriness”. “Frustration” could absorb some of the functionalities that have now been included in “Self-Confidence”, namely those related with the negative situations, i.e. those that promote a decrease in “Self-confidence”. This would make our model cleaner by avoiding the use of bi-directional Emotional Mechanisms, whose Emotional Accumulators are both increased and decreased by corresponding Emotional Evaluation Functions. Another interesting Emotional Mechanism that could lead to good results is “Angriness”, which is usually related to blocking the environment’s negative effects. “Angriness” is important in motivating the Agent to maintain a consistent (“aggressive”) behavior towards the environment, even when the situations are clearly negative. This could be useful in the discovery of sudden environment drifts.

Supplementary work on Emotional Mechanisms may be developed if we choose to enlarge the scope of such Mechanisms to Agent Coordination. If we consider teams of Agents instead of individual Agents, what will be the function of Emotional-Mechanisms? We have already explained that certain Emotions could be useful in the context of team coordination. For example, “Self-Confidence” is important in promoting Goal revision and, if applied to the decisions of a team coordinator, could involve moving an entire team from its current fire fighting positioning to another possibly less dangerous fire-front. However, how this and other Emotional Mechanisms could be used to enable more subtle tactical variations remains an unanswered question.

Finally, it would also be interesting to address some issues regarding *Personality*. If we consider that *Personality* is the set of possible behavior variations among Agents sharing the same Architecture, it seems easy to implement this notion in the context of Emotional Agents. In fact, as shown in Chapter 4, we included several parameters in the model of Emotional Accumulators. These parameters are able to control how fast a given Emotional Mechanism is elicited and how long it remains active.

By changing these parameters, it is possible to obtain Agents that reach higher levels of “Fear” or “Self-Confidence” faster than others, enabling the emergence of different *Personalities* amongst Agents using the exact same Architecture. Many questions will immediately follow. For example, is there an optimal *Agent Personality* to cope with a given environment? Alternatively: is there an optimal combination of Agent *Personalities* resulting in efficient teams (for a given set of possible environment variations)?

These are all very interesting questions for future work.

REFERENCES

[Bar99] Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36:253–263, 1999. http://citeseer.nj.nec.com/bartlett99measuring.html.

[Bat97] Joseph Bates. The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125, 1997. http://citeseer.nj.nec.com/bates94role.html.

[Bea94] Luc Beaudoin. *Goal processing in autonomous agents*. PhD thesis, School of Computer Science, The University of Birmingham, 1994. http://citeseer.nj.nec.com/beaudoin94goal.html.

[Bot01] Luis Botelho and Hélder Coelho. Machinery for artificial emotions. *Journal of Cybernetics and Systems*, 32(5):465–506, 2001. http://taylorandfrancis.metapress.com/ index/M9P21AW7TYY4P97F.pdf.

[Bot95] Luis Botelho and Hélder Coelho. A schema-associative model of memory. In *Proceedings of the 4th Golden West International Conference on Intelligent Systems*, pages 81–85, S. Francisco, CA, USA, 1995. http://citeseer.nj.nec.com/context/2256534/0.

[Bow00] Gordon H. Bower and Joseph P. Forgas. Affect, memory, and social cognition. In Eric Eich, editor, *Cognition and Emotion*. Oxford University Press, 2000.

[Cah90] Janet E. Cahn. The generation of affect in synthesized speech. *Journal of the American Voice I/O Society*, July 1990. http://citeseer.nj.nec.com/article/janet90generation.html.

[Can00] Lola Cañamero. Designing emotions for activity selection. Technical Report TR - DAIMI PB 546, Univesity of Arhaus, 2000. http://citeseer.nj.nec.com/384344.html.

[Can97] Lola Cañamero. A hormonal model of emotions for behavior control. Presented as poster at the Fourth European Conference on Artificial Life (ECAL '97), Brighton, UK, 1997. http://citeseer.nj.nec.com/canamero97hormonal.html.

[Can98] Lola Cañamero. Issues in the design of emotional agents. In *AAAI Fall Symposium on Emotional and Intelligent: The tangled knot of cognition*, pages 49–54, Menlo Park, CA, 1998. http://citeseer.nj.nec.com/437878.html.

[Cas00] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill (editors). *Embodied Conversational Agents*. MIT Press, 2000.

[Cos00] Monica Costa, Liwei Zhao, Diane M. Chi, and Norman I. Badler. The emote model for effort and shape. In *Proceedings of SIGGRAPH 2000*, pages 173–182, 2000. http://citeseer.nj.nec.com/costa00emote.html.

[Cus99] Luis Custódio, Rodrigo Ventura and Carlos Pinto-Ferreira. Artificial emotions and emotion-based control systems. In *Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation*, 1999. http://citeseer.nj.nec.com/308020.html.

[Dam94] Antonio R. Damasio. *Descartes Error - Emotion, Reason and the Human Brain*. Gosset/Putnam Press, NY, 1994.

[Dyn] Boston Dynamics. Peopleshop. http://www.bdi.com/content/sec.php?section=newsarticle&article=12.

[Ekm94] Paul Ekman and Richard J. Davidson (editors). *The Nature of Emotion - Fundamental Questions*. Oxford University Press, 1994.

[Ell99] Clark Elliott, Jeff Rickel, and James Lester. Lifelike pedagogical agents and affective computing: An exploratory synthesis. *Lecture Notes in Computer Science*, 1600:195–212, 1999. http://citeseer.ist.psu.edu/elliott99lifelike.html.

[Fie00] Klaus Fiedler and Herbert Bless. The formation of beliefs at the interface of affective and cognitive processes. In Nico H. Frijda, Antony S. R. Manstead, and Sacha Bem, editors, *Emotions and Beliefs*. Cambridge University Press, 2000.

[Fle02] Michael Fleischman and Eduard Hovy. Towards emotional variation in speech-based natural language generation. In *Proceedings of the Second International Natural Language Generation Conference (INLG'02)*, New York, USA, 2002. http://citeseer.nj.nec.com/541400.html.

[Fri86] Nico H. Frijda. *The Emotions*. Cambridge University Press, 1986.

[Fri94] Nico H. Frijda. Emotions are functional, most of the time. In Paul Ekma and Richard J. Davidson, editors, *The Nature of Emotion - Fundamental Questions*. Oxford University Press, 1994.

[Gad02] Sandra C. Gadanho. Emotional and cognitive adaptation in real environments. In *the symposium ACE'2002 of the 16th European Meeting on Cybernetics and Systems Research*, Vienna, Austria, 2002. http://citeseer.ist.psu.edu/488817.html.

[Gad98] Sandra C. Gadanho and Jonh Hallam. Emotion-triggered learning for autonomous robots. In Dolores Cañanero, Chisato Numaoka, and Paollo Petta, editors, *SAB '98 Workshop on Grounding Emotions in Adaptive Systems*, pages 31-–36, August 1998. http://www.dai.ed.ac.uk/papers/documents/rp916.html.

[Gra00] Jonathan Gratch. Émile: Marshalling passions in training and education. In Carles Sierra, Maria Gini, and Jeffrey S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 325–332, Barcelona, Catalonia, Spain, 2000. http://citeseer.ist.psu.edu/gratch00emile.html.

[Gra01] Jonathan Gratch and Stacy Marsella. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 278–285, Montreal, Canada, 2001. http://citeseer.nj.nec.com/gratch01tears.html.

[Kli99] Christopher Kline and Bruce Blumberg. The art and science of synthetic character design. In *Proceedings of the AISB 1999 Symposium on AI and Creativity in Entertainment and Visual Art*, Edinburgh, Scotland, 1999. http://citeseer.ist.psu.edu/kline99art.html.

[Led96] Joseph Ledoux. *The Emotional Brain: The Mysterious Underpinnings of Emotional Life*. Simon & Schuster, New York, 1996.

[Mar02] Stacy Marsella and Jonathan Gratch. A step toward irrationality: using emotion to change belief. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '02)*, pages 334–341, Bologna, Italy, 2002. http://www.ict.usc.edu/publications/agents02-coping.pdf.

[Mau93] Calvin R. Maurer and J. Michael Fitzpatrick. A review of medical image registration. In R. J. Maciunas, editor, *Interactive ImageGuided Neurosurgery*, pages 17–44, Park Ridge, IL, 1993. American Association of Neurological Surgeons. http://citeseer.ist.psu.edu/maurer93review.html.

[Mic04] Microsoft Corporation. Microsoft agent. http://www.microsoft.com/msagent/default.asp.

[Mof00] Dave Moffat and Nico Frijda. Functional models of emotion. In G. Hatano, N. Okada, and H. Tanabe, editors, *13th Toyota Conference – Affective-Minds*, pages 169–181, Elsevier, Amsterdam, 2000.

[Mou03] Daniel C. Moura. Emotional Agents. Technical Report, Laboratório de Inteligência Artificial e Ciências dos Computadores, Universidade do Porto, 2003. http://www.fe.up.pt./~dcm/id/emotional_agents.pdf

[Oli02] Eugénio Oliveira and Luís Sarmento. Emotional valence-based mechanisms and agent personality. In Guilherme Bittencourt and Geber Ramalho, editors, *Advances in Artificial Intelligence, Proceedings of SBIA02*, volume 2507 of *Lecture Notes in Computer Science*, Brazil, 2002. Springer. http://www.fe.up.pt/ eol/PUBLICATIONS/2002/personality7.ps.

[Oli03] Eugénio Oliveira and Luís Sarmento. Emotional advantage for adaptability and autonomy. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS'03)*, pages 305–312, New York, USA, 2003. ACM Press. http://doi.acm.org/10.1145/860575.860625.

[Ort01] Andrew Ortony. On making believable emotional agents believable. In Robert Trappl, Paolo Petta, and Sabine Payr, editors, *Emotions in Humans and Artifacts*. MIT Press, Cambridge, 2001.

[Ort88] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[Ouc02] Erik Ouchterlony and Ragnar Ouchterlony. Real time battle simulator, 2002. http://realtimebattle.sourceforge.net/.

[Pfe93] Rolf Pfeifer. Studying emotions: Fungus eaters. In *Proceedings of the 2nd European Conference on Artificial Life*, pages 916–927. Université Libre de Bruxelles, 1993.

[Pic97] Rosalind W. Picard. *Affective Computing*. MIT Press, 1997.

[Piw02] Paul Piwek. An annotated bibliography of affective natural language generation. Technical Report ITRI-02-02, University of Brighton, 2002. http://www.itri.brighton.ac.uk/ Paul.Piwek/affect-bib.pdf.

[Roh93] Warren M. Rohsenow, James P. Hartnett, and Young I. Cho (Editors). *Handbook of Heat Transfer*. McGraw-Hill Professional, 3rd edition, 1993.

[Rol99] Edmund T. Rolls. *The Brain and Emotion*. Oxford University Press, 1999.

[Ros99] Fiorella de Rosis and Floriana Grasso. Affective natural language generation. In *International Workshop on Affective Interactions (IWAI '99)*, pages 204–218, 1999. http://citeseer.nj.nec.com/357169.html.

[Rus95] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[Rus99] Kenneth E. Russell and Bruce M. Blumberg. Behaviour friendly graphics. In *Computer Graphics International conference (CGI'99)*, Canmore, Alberta, Canada, 1999.

[Sch00] Norbert Schwarz. Emotion, cognition, and decision making. *Cognition and Emotion*, 14(4):434–606, 2000.

[Sil02] Barry G. Silverman, Michael Johns, Ransom Weaver, Kevin O'Brien, and Rachel Silverman. Human behavior models for game-theoretic agents: Case of crowd tipping. To apper in CogSci Quarterly, 2002. http://citeseer.nj.nec.com/silverman02human.html.

[Sim67] Herbert Simon. Motivational and emotional controls of cognition. *Psychological Review*, 74(1):29–39, 1967.

[Slo00] Aaron Sloman and Brian Logan. Evolvable architectures for human-like minds. In G. Hatano, N. Okada, and H. Tanabe, editors, *13th Toyota Conference – Affective-Minds*, pages 169–-181, Elsevier, Amsterdam, 2000. http://citeseer.ist.psu.edu/sloman00evolvable.html.

[Slo01] Aaron Sloman. Beyond shallow models of emotion. *Cognitive Processing*, 2(1):177–198, 2001. http://www.cs.bham.ac.uk/research/cogaff/.

[Slo81] Aaron Sloman and Monica Croucher. Why robots will have emotions. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 197–202, Vancouver, 1981. http://www.cs.bham.ac.uk/research/cogaff/.

[Slo98] Aaron Sloman. Damasio, descartes, alarms, and metamanagement. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '98)*, pages 2652–7, San Diego, CA, USA, 1998. http://citeseer.ist.psu.edu/sloman98damasio.html.

[Vel98a] Juan Velásquez. Modeling emotion-based decision-making. In Dolores Cañanero, editor, *Proceedings of the 1998 AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*, pages 164–169, Orlando, FL, USA, 1998. http://www.ai.mit.edu/people/jvelas/papers/Velasquez-FS98.ps.

[Vel98b] Juan Velásquez. A computational framework for emotion-based control. In *Proceedings of the Grounding Emotions in Adaptive Systems workshop, SAB '98*, Zurich, Switzerland, 1998.

[Ven00] Rodrigo Ventura. Emotion-based agents. Master's thesis, Instituto Superior Técnico, Lisboa, Portugal, 2000.

[Wri97] Ian Wright. *Emotional Agents*. PhD thesis, School of Computer Science, The University of Birmingham, 1997. http://www.cs.bham.ac.uk/research/cogaff/.

[Yoo00] Song-Yee Yoon, Robert C. Burke, Bruce Blumberg, and Gerald E. Schneider. Interactive training for synthetic characters. In *Proceedings of AAAI/IAAI 2000*, pages 249–254, 2000. http://citeseer.ist.psu.edu/yoon00interactive.html.

[Zha03] Wen-Yi Zhao, Rama Chellappa, P.J. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Computing Survey*, pages 399–458, December 2003. http://citeseer.nj.nec.com/zhao00face.html.

[Zho03] Xiaoming Zhou and Cristina Conati. Inferring user goals from personality and behavior in a causal model of user affect. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '03)*, pages 211–218, Miami, FL, USA, 2003. http://www.cs.ubc.ca/~conati/my-papers/iui03-camera.pdf.

APPENDIX A. THE MATHEMATICAL MODEL OF PYROSIM

A.1. Introduction

In this Appendix we will present the model developed to build the Pyrosim Platform.

We will start by describing a fire-fighting environment and try to extract the main features that need to be modeled in order to develop a believable simulation. Pyrosim is not intended to be a realistic simulation in the sense that it will run as close as possible to the real world. Instead, we are only interested in replicating the reality approximately, just enough to create situations with similar complex requirements. For this reason, we do not need to develop very complex models. A high degree of complexity could even be problematic because the consequent computational effort probably would not allow the simulation to run in real-time for reasonably sized scenarios.

We will then present the derivation of the fire propagation and extinction models. Many simplifications will be made in order to develop simple models. Nevertheless, the propagation model incorporates the influence of vegetation, terrain geometry and wind. Next, we will present the model that supports the calculation of Agent mobility. Agents are able to control their acceleration and direction while suffering from constraints imposed by terrain geometry.

A.2. Some facts about forest fires

A.2.1. The Process of Ignition

The process of ignition of a combustible solid material is a complex phenomenon, involving both chemical and physical reactions. In a first state, it involves the chemical decomposition of the material surface into flammable volatiles that are then ignited and start the actual surface combustion. These volatiles are released from the solid's surface when the temperature increases, however, they only ignite when their temperature reaches a certain threshold point – the flashpoint, T_f. The ignition of these volatiles will increase the solid's temperature that will eventually ignite if it reaches another threshold temperature known as the fire point, T_{ig}. Depending on its nature, the solid will possibly attain a sustained combustion state after ignition.

The temperature increase in the solid's surface, which leads to the ignition of the volatiles, may have three different causes that give rise to three forms of ignition. Namely:

1. *Pilot Ignition*: due to the presence of a pilot flame;
2. *Spontaneous Ignition*: by heat transference from the atmosphere by effect of radiation or convection;
3. *Surface Ignition*: by a combination of the two previous factors.

In forests, fire-fronts develop and spread through Surface Ignition. Vegetation immediately ahead of the fire-front is ignited not only because of direct exposure to the fire-front itself, but also due to the high temperatures in its surroundings, which may reach values above 500ºC within a few meters.

On the other hand, fire usually starts by the spontaneous ignition of certain fuel elements in the forest with lower fire points. This happens frequently in dry seasons when the reduction of moisture in certain organic elements covering the ground may decrease their flashpoint / fire point temperature, thus increasing the probability of spontaneous ignition.

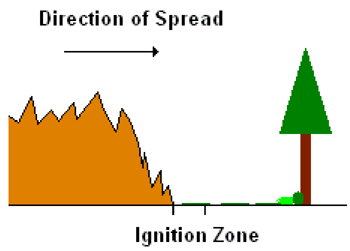

Figure A. 1 The typical progression of a fire-front.

A.2.2. The Spreading of the Fire-Front

As seen in the previous section, the spreading of a fire-front is based on the surface ignition of fuel elements in its close surroundings. Despite this apparently simple process, the overall fire-front behavior results from a complex interaction between vegetation, topography and meteorological conditions. Let us briefly analyze each one of these items.

A.2.2.1. Vegetation

Vegetation is the major source of unpredictability in fire spreading. It introduces several variables whose value and impact are very difficult or impossible to determine. For instance, during their life span, plants are exposed to stress caused by biological agents and meteorological conditions (draught, wind…), which greatly influence both their resistance and behavior towards fire. Other important factors include soil conditions, the growth rate, the amount of dead material present in the plant as well as on the ground, and also the existence of physical damage.

When considering the fire behavior of a particular plant, it is important to take into account six basic variables in order to correctly understand the spreading of fire in a forest:

1. **Moisture.** Water is the most important element in preventing combustion of organic elements because it acts as an effective heat sink. With increasing temperatures, the water in the plants will volatilize at a rate that depends inversely on the square of the size of plants constituents. A branch with 6 mm will lose water twice as fast as a 12 mm branch.
2. **Extractives.** Extractives inside a plant present a wide variety of possible fire behaviors. Some extractives in plants will volatilize at temperatures inferior to that of the moisture while others will only volatilize near the point of thermal degradation of wood. The combination of moisture and some extractives produces an interesting fire delaying mechanism that diminishes the combustion process. However, when the vegetation is substantially dry, extractives can contribute significantly to combustion by producing 30% more heat than the heat produced by dry wood itself.
3. **Foliage.** Foliage is the most fire critical component of the plant. In fact, the thin material that composes most foliage favors a very quick loss of moisture, making foliage a fast fuel. Dry or dead foliage is especially flammable, contributing to a rapid spread of flame to the rest of the plant and, eventually, to other plants.
4. **Twigs and Branches.** Both twigs and branches, depending on their width and volume may represent a significant portion of the fuel. When they are still green they do not contribute very much to the combustion but dry or dead twigs and branches may burn with intermediate intensity.
5. **Stem.** The main stem is usually the major part of the plant's biomass. It is quite resistant to flash combustions, as the ones resulting from burning foliage or grass. On the other hand, when ignited, they provide a large amount of fuel that will burn continuously for a long period.

Generically, plants with the following characteristics will favor a rapid spread of fire:

- High surface to volume ratio;
- Low moisture content;
- Presence of a high percentage of dead material.

Dry grass and foliage are, therefore, extremely dangerous in fire propagation.

A.2.2.2. Topography

It is a well-know fact that fire propagates faster in the upward direction. Therefore, the local topography deeply influences the spreading of the fire-front: fire will tend to advance in the direction of positive slope, climbing up hills faster than it proceeds over flat regions (S1 > S2).

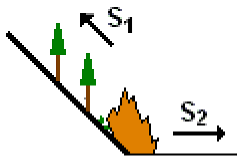

Figure A. 2. The speed of the climbing fire-front (S1) is higher than the speed of the fire-front progressing in the flat area (S1).

The exact relation between the terrain slope and the speed of spread of fire is difficult, if not impossible, to obtain. In spite of this, laboratory tests may help us develop a finer picture of this relation. The following table presents some laboratory results about the speed of spread of fire in strips of filter paper [Roh93].

Table A. 1. Spreading Speed as a function of the burning surface slope. Values refer to burning strips of filter paper.

Orientation (º)	Spreading Speed (m/s)
0	3.6
22.5	6.3
45	11.2
75	29.2
90	46-74

Although not shown in Table 1, it is interesting to note that for negative slope values (downward direction) below -30º the speed of spread reaches a down limit corresponding to a slow, yet stable, spreading. Considering the entire slope range (-90º, 90) the expected speed of spread would be something similar to the exponential curve presented in Figure 3. The values presented refer to the spread over strips of filter paper.

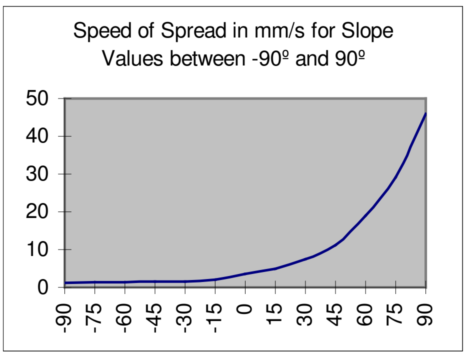

Figure A. 3. The speed of spread of the fire-front over the entire slope range (-90 to +90).

A.2.2.3. Meteorological Conditions

Weather conditions have direct impact on the spread of fire. As far as the ignition process is concerned, air temperature and relative humidity are crucial factors since they have direct impact on fire point temperature. Thus, high air temperatures, combined with reduced humidity, will favor a rapid spread of fire. Once again, this relation is very difficult to establish and depends heavily on the existent type of vegetation. For analysis purposes, and despite their fundamental role in the fire process, these factors are usually greatly simplified.

Another very important factor is wind influence. Besides providing transportation for burning or incandescent elements (which may start fires in nearby locations), wind dramatically alters the fire-front's speed and direction. When blowing from behind the fire-front, the wind will enrich the combustion process. This will increase heat production and its transfer to the region immediately ahead contributing to greater fire spreading speeds.

On the other hand, if the wind velocity reaches certain high speeds, the heat around the fire-front will be dissipated, significantly slowing down the speed of spread or even extinguishing the flame. Figure 4 shows the typical spreading behavior of a fire-front for increasing wind speeds.

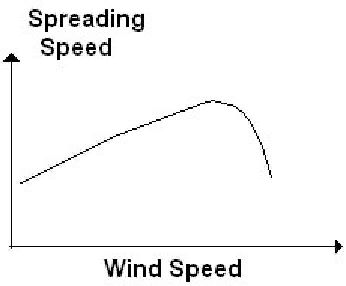

Figure A. 4. Spreading Speed Vs. Wind Speed

A.2.3. The Point of Extinction

Extinction is basically the inverse process of ignition, again involving two critical temperature thresholds. By cooling a flame below the fire point temperature, the essential condition for a sustained combustion is suppressed and the flame will go out. The flammable vapors, however, will continue to be expelled, keeping the possibility of a later re-ignition. By further reducing the temperature it is possible to stop the release of the flammable volatiles, and thus effectively prevent the ignition of a new flame.

There are several ways to achieve cooling. When fighting forest fires, the most frequent cooling process consists in throwing water over the burning surface. The water will cool the burning surface by absorbing its heat during its transformation into vapor. At 25ºC, water can absorb 2.4 kJ/g of energy in the evaporation process. The rise of the resulting vapor into the atmosphere will also favor the dispersion of heat.

Using water to fight fire has important advantages over the use of chemical suppressants. Besides being much more economical, water does not pose any threat to the fire fighting personnel nor to the environment. This is very relevant as some highly effective chemical suppressants often create extremely dangerous atmospheres for humans, which can greatly complicate firefighting procedures.

A.3. Pyrosim Simulator

All the simulation in Pyrosim is made within a quadrangular grid containing N X N cells. Cells are the simulation's basic entities and all its content is considered homogeneous along the cell. Each cell has its own atmosphere temperature and moisture variables. A cell may be populated by several types of vegetation elements and also by underground fuel. Fire propagates from cell to cell according to a process that will be described later. Figure 5 tries to illustrate a Pyrosim cell in a simplified fashion.

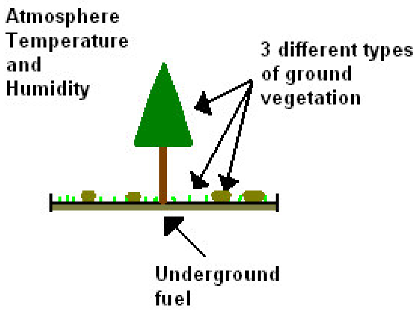

Figure A. 5. The basic elements of a Pyrosim cell.

A.3.1. Topography

The N x N grid used in Pyrosim takes into account the information about the relative altitude of each cell. In this way, the overall grid may represent realist regions that include all types of topographic elements. The altitude information is essential to simulate different speeds of fire spreading, as described before. Currently, Pyrosim does not support data from GIS. Instead, and for now, it generates its own topography according to user parameters. Figure 6 shows a 3D representation of a terrain generated by Pyrosim.

Figure A. 6. An example terrain generated by Pyrosim

A.3.2. Vegetation

As explained in the previous section, vegetation takes the leading role in the fire-front progression. To produce a realistic simulation, with an interesting set of fire behaviors, several types of vegetation must be considered. As we have seen, different vegetation species show distinct burning properties. Pyrosim explores:

- **Temperature of Ignition** – the temperature at which the vegetation will start burning: TMAX (ªC).
- **Burning Rate** – the amount of heat produced at each second by the burning plant: BR (kJ/s).
- **Overall Energy** – the overall energy that a vegetation element has to burn initially: ETOTAL (J).

A Pyrosim cell supports simultaneously four types of vegetation: underground vegetation, grass, twigs and trees. Table 2 presents a brief comparison between the mentioned types of vegetation in respect to the three burning properties listed before.

Table A. 2. Vegetation types in Pyrosim and its basic properties.

	T_{MAX}	BR	E_{TOTAL}
Underground	Medium	Low	Medium
Grass	Low	High	Low
Twigs	Medium	Medium	Medium
Trees	High	Medium	High

Each vegetation element has two possible states: burning or not burning. Consider a vegetation element VEG(i) with a given $T_{MAX}(i)$, BR(i) and $E_{TOTAL}(i)$. Let T_{ENV} be the environment temperature of the corresponding cell and let $E_{BURNED}(i)$ be the amount of energy already consumed by the fire. The Ignition Rule used by Pyrosim is:

IF (T_{ENV} >= $T_{MAX}(i)$) AND ($E_{BURNED}(i) < E_{TOTAL}(i)$) THEN IGNITE VEG(i)

Extinction of a ignited vegetation element VEG(i) can happen according to the Extinction Rule:

IF ($T_{ENV} < T_{MAX}(i)$) OR ($E_{BURNED}(i)$ >= $E_{TOTAL}(i)$) THEN EXTINCT VEG(i)

During an interval of Δt seconds of combustion, a vegetation element will transfer to the environment an amount of energy given by:

$$E_{PRODUCEDd}(i,\Delta t) = BR(i) * \Delta t$$

Equation 1

The energy produced will increase the cell's global temperature. The properties on table 2 suggest that the predictable spreading behavior of the fire will possibly be:

- With the rising of the cell temperature, grass will be the first vegetation element to start burning.
- The heat produced may ignite both underground vegetation and twigs.
- Eventually, the heat being produced at that moment will ignite the trees, which have a huge amount of energy to consume. During this period, it is possible that grass and twigs have been completely consumed and have now stopped burning (because of their small overall energy and high burning rate).

The ignition temperature of vegetation elements may change according to the supporting cell's moisture. Dryer cells will naturally promote lower ignition temperatures, whereas highly moisturized ones will make ignition temperatures of vegetation elements rise.

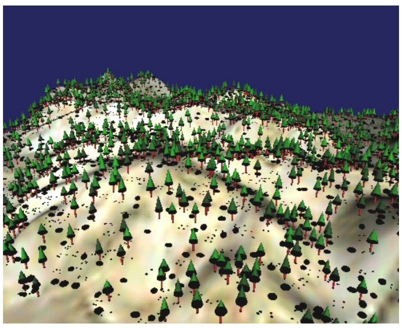

Figure A. 7. A close look over a Pyrosim terrain showing the generated vegetation.

A.3.3. Fire Propagation

In Pyrosim, fire propagation between cells takes place only by means of heat exchange. That is, it is the diffusion of heat from the burning cell to the neighbor cells that may ignite one of these. Therefore, most of the modeling effort of Pyrosim is done over this single issue: heat transfer. There are several issues that must be considered, namely:

- Heat Production and Vegetation Type
- Heat Transfer and Dissipation
- Influence of Topography
- Wind Influence

A.3.3.1. Heat Production

In Pyrosim, vegetation combustion is the only source of heat. We have already seen that each cell may have up to four vegetation elements, one of each supported types (refer to table 2). Consider a vegetation element VEG(i) with defined T_{MAX}(i), BR(i) and E_{TOTAL}(i). The production of heat and the corresponding rise of cell temperature, ΔT_{CELL}, can be described by the following cycle:

WHILE VEG(i) isBurning

{

$E_{PRODUCED}(i,Dt) = BR(i) \times \Delta t$

$\Delta T_{CELL} = E_{PRODUCED}(i, \Delta t) \times K_{AIR}$

$T_{CELL} = T_{CELL} + \Delta T_{CELL}$

}

For each burning vegetation element in the cell a similar cycle is performed. K_{AIR} is a constant that relates the increase of air temperature in the cell with the amount of energy released during combustion. K_{AIR} is expressed in ºC/J.

A.3.3.2. Heat Transfer and Dissipation

During combustion, a portion of the heat produced in a cell is transferred to the neighbor cells while the other portion is dissipated to the environment. Figure 8 depicts these processes.

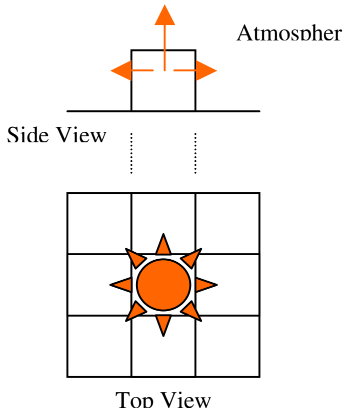

Figure A. 8. A Pyrosim cell releases heat to the atmosphere and exchanges heat with its 8 neighbors.

The heat dissipated to the environment, at temperature T_{ENV}, will decrease the cell's global temperature, T'_{CELL}, in accordance to the following equations:

$$\frac{\partial T'_{CELL}}{\partial t} = -(T'_{CELL} - T_{ENV}) \times D_A$$

Equation 2

$$T'_{CELL}(t) = T_{ENV} + (T'_{CELL}(0) - T_{ENV}) \times e^{-D_A t}$$

Equation 3

DA is the Dissipation Factor ($0 > D_A > 1$) that relates the amount of heat lost by the cell (at temperature T'_{CELL}) to the environment, assuming a constant temperature of T_{ENV}. Pyrosim will update values at discrete time intervals as frequently as possible. However, update intervals will depend on the current load of the simulator and, of course, on the machine. The updated value of the cell temperature T'_{CELL}, for an interval of Δt seconds after the last update will be:

$$T_{CELL} = T'_{CELL}(\Delta t) = T_{ENV} + (T'_{CELL}(0) - T_{ENV}) \times e^{-D_A \Delta t}$$

Equation 4

Part of the resulting heat, T_{CELL}, is transferred to the neighbor cells. The amount of heat transferred between two cells is proportional to the temperature difference between the source and target cells. Let the temperature of the source cell be T_{SOURCE} and the temperature of the target cell be T_{TARGET}. The heat transfer will occur according to:

$$\frac{\partial T_{SOURCE}}{\partial t} = -(T_{SOURCE} - T_{TARGET}) \times K_{Tranf}$$

Equation 5

Alternatively:

$$\frac{\partial^2 T_{SOURCE}}{(\partial t)^2} = -(\frac{\partial T_{SOURCE}}{\partial t} - \frac{\partial T_{TARGET}}{\partial t}) \times K_{Tranf}$$

Equation 6

But, since there are energy losses:

$$\frac{\partial T_{SOURCE}}{\partial t} = -\frac{\partial T_{TARGET}}{\partial t}$$

Equation 7

Then:

$$\frac{\partial^2 T_{SOURCE}}{(\partial t)^2} = -2 \times (\frac{\partial T_{SOURCE}}{\partial t}) \times K_{Tranf}$$

Equation 8

Finally:

$$T_{SOURCE}(t) =$$
$$\frac{1}{2}(T_{SOURCE}(0) \times (1 + e^{-2K_{Transf}t}) + T_{TARGET}(0) \times (1 - e^{-2K_{Transf}t})$$

Equation 9

Pyrosim will simulate the heat transfer between cells during a time interval of Δt by reducing the value of the source cell temperature and increasing the target cell temperature by an amount of ΔT:

$$\Delta T(\Delta t)_{(SOURCE,TARGET)} = T_{SOURCE}(\Delta t) - T_{SOURCE}(0) =$$
$$(e^{-2K_{Transf}\Delta t} - 1) \times (T_{SOURCE}(0) - T_{TARGET}(0))$$

Equation 10

K_T is the transference factor that, as will be seen later, depends on several parameters, including the terrain topography. For each cell, there are eight different ΔT terms that must be computed, one for each neighbor cell.

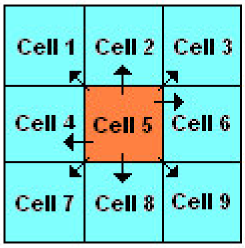

Figure A. 9. A 3x3 mask and the corresponding cell numbering.

Pyrosim will iterate across the entire terrain grid using a 3X3 mask. The mask's central cell will be updated by adding the contributions of the corresponding eight neighbor cells (refer to figure 9). Note that during this process each pair of neighbor cells will be accounted twice (once as source cell, and once as target cell), so we must divide the ΔT's calculated by 2. Temperature variation in the central cell, from iteration k to k+1 (with an update interval of Δt seconds) will be:

$$\Delta T_5^{k+1}(\Delta t) = \frac{1}{2}\sum_{\substack{i=1\\ i\neq 5}}^{9} \Delta T_{(5,i)}(\Delta t)$$

Equation 11

The updated temperature of the central cell will be given by:

$$T_5^{k+1} = T_5^k + \Delta T_5(\Delta t)$$

Equation 12

A.3.3.3. Influence of Topography

As seen previously, heat transfer between two adjacent cells depends on their relative slope. We must, therefore, include the influence of slope in the equation (d). This can be achieved by decomposing the transference factor K_{Transf} in two more specific factors:

$$K_{Transf} = K_{TCELL} \times F_{SLOPE}(\theta)$$

Equation 13

K_{TCELL} is a constant that determines the transference factor between two adjacent cells, in the horizontal plane. F_{SLOPE} is a function with the following properties:

$$F_{SLOPE}(0) = 1$$

$$F_{SLOPE}(\theta_1) > F_{SLOPE}(\theta_1) : \theta_1 > \theta_2$$

Equation 14

Pyrosim uses the following exponential function extracted from the data presented in figure 3:

$$F_{SLOPE}(\theta) = [K_{SLOPE}]^{\theta}$$

Equation 15

with K_{SLOPE} = 1.025.

Table 3 shows some sampled values.

Table A. 3. The FSLOPE(θ) function.

θ (º)	$F_{SLOPE}(\theta)$	θ (º)	$F_{SLOPE}(\theta)$
-90	0,11	0	1,00
-75	0,16	15	1,45
-60	0,23	30	2,10
-45	0,33	45	3,04
-30	0,48	60	4,40
-15	0,69	75	6,37
0	1,00	90	9,23

The value of K_{SLOPE} may be adjusted, if needed. Reducing the value will flatten the exponential curve so that all values of the function will tend to $F_{SLOPE}(0º)$.

Let us, for now, consider the following, rather simplified, situation:

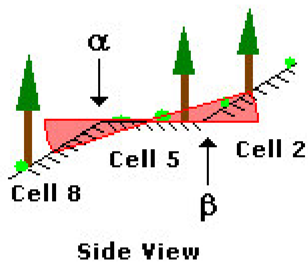

Figure A. 10. The profile view of the 3 consecutive cells. α is a negative angle while β is positive.

Using equations 9 and 12, the heat transfer from the central cell (5) to its neighbors (cells 8 and 2), including the influence of topology, will be given by:

$$\Delta T_{(5,8)}(\Delta t) = (e^{-2\times K_{TCELL}\times F_{SLOPE}(\alpha)\times\Delta t} - 1)\times(T_5(0) - T_8(0))$$

$$\Delta T_{(5,2)}(\Delta t) = (e^{-2\times K_{TCELL}\times F_{SLOPE}(\alpha)\times\Delta t} - 1)\times(T_5(0) - T_2(0))$$

Note that the angles between cells are calculated by tracing a line crossing the centers of the corresponding cells. α is a negative angle while β is positive.

In a more generic situation, we will have a 3X3 cell grid and each cell may have an arbitrary slope. Figure 11 depicts a situation where the angles established between cells create a more complex heat transfer process.

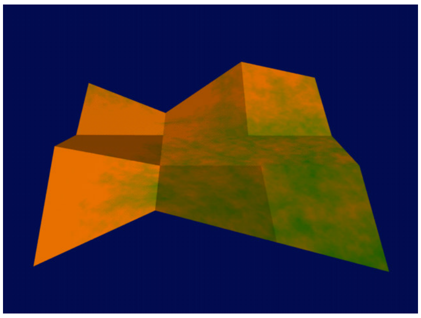

Figure A. 11. In most cases, Pyrosim cells will have complex slope relations.

Pyrosim will calculate the eight angles corresponding to the eight neighbor cells using the same procedure: the angle between two cells is calculated by connecting the centers of the corresponding cells.

As before, heat transfer will then be determined using equations 9 and 12 for the eight cells with the appropriate values of the angles calculated.

A.3.3.4. The Influence of Wind

In the presence of wind, heat transfer between cells may be increased or decreased depending on the direction of the wind. The next figure shows a situation where a 3 X 3 grid is under the influence of wind blowing with intensity W (m/s) and direction θ:

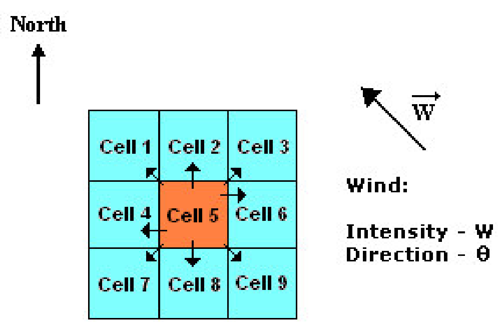

Figure A. 12. Temperature exchange under the influence of wind.

Pyrosim takes into account the influence of wind by including the term T_W in the value of K_{Transf} :

$$T_{W(SOURCE,TARGET)} = K_W \times W \times \cos(\alpha_{WST})$$

Equation 16

where:

- K_W - is the Wind Influence Constant.
- W - is the Wind Intensity.
- α_{WST} – is the angle between the wind direction and the horizontal line that connects the centers of source and target cells.

The complete K_{Transf} value, including both the influence of wind and topography will be given by:

$$K_{Transf} = K_{TCELL} \times F_{SLOPE}(\theta) + K_W \times W \times \cos(\alpha_{WST})$$

Equation 17

By using the T_W term we make sure that when the wind is blowing in the opposite direction of a heat transfer vector, the amount of heat exchange is reduced, as one would expect, since $\cos(\alpha_{WST})$ will have a negative value.

A.3.4. Revision of Heat/Temperature Exchange Equations

We will now summarize the complete set of equations we have been deriving. We are assuming an update interval of Δt seconds for all the equations.

Firstly, Pyrosim must take into account the dissipation of heat to the atmosphere.

$$T'_{CELL}(t) = T_{ENV} + (T'_{CELL}(0) - T_{ENV}) \times e^{-D_A t}$$

Equation 18

The next step is to update the cell temperature to include the effect of heat exchange with its eight neighbors. For a single neighbor, the temperature change is determined by the following equation:

$$\Delta T(\Delta t)_{(SOURCE,TARGET)} = T_{SOURCE}(\Delta t) - T_{SOURCE}(0) = \\ (e^{-2K_{Transf}\Delta t} - 1) \times (T_{SOURCE}(0) - T_{TARGET}(0))$$

Equation 19

The influence of topography and wind is hidden in the transference factor K_{Transf}. Expanding K_{Transf}:

$$K_{Transf} = K_{TCELL} \times [K_{SLOPE}]^{\theta_{ST}} + K_W \times W \times \cos(\alpha_{WST})$$

Equation 20

with,

- K_{TCELL}: temperature transfer constant between cells in the horizontal plane.
- K_{SLOPE}: slope effect constant.
- θ_{ST}: slope angle between source and target cell.
- $T_{SOURCE}(0)$: temperature of the source cell, after the last iteration.
- $T_{TARGET}(0)$: temperature of the target cell, after the last iteration.
- K_W: wind constant.
- W: wind intensity.
- α_{WST}: instant angle between the direction of wind and heat transfer vector.

For a 3X3 grid, we must consider the temperature exchange between the central cell and its eight neighbors.

$$\Delta T_5^{k+1}(\Delta t) = \frac{1}{2} \sum_{\substack{i=1 \\ i \neq 5}}^{9} \Delta T_{(5,i)}(\Delta t)$$

Equation 21

Finally, the resulting temperature in the cell will be:

$$T_5^{k+1} = T_5^k + \Delta T_5(\Delta t)$$

Equation 22

A.3.5. The Extinction Process in Pyrosim

The extinction of an ignited cell can happen in two different situations:

1. The energy available in the cell has been totally consumed by combustion.
2. The temperature of the ignited cell has dropped below the fire point temperature (T_{MAX}).

We have already seen in a previous section that water is very effective in fighting fire, since it works as a powerful heat sink. Water thrown over a combusting element will absorb part of its heat and evaporate in the atmosphere. Thereby, the temperature of the burning element will be decreased, eventually below the fire point.

Let us define H_{EVAP} as the amount of heat absorbed by a volume of water V_{WATER} thrown over an ignited Pyrosim cell, during the time interval Δt:

$$H_{EVAP} = V_{WATER} \times KA_{WATER}$$

Equation 23

KA_{WATER} is the *Constant of Absorption* of water and relates the amount of energy absorbed with the volume of water evaporated. At 25ºC, the value of KA_{WATER} will be approximately 2.4 x 106 J/liter.

Almost certainly, not all water thrown over a burning element will evaporate: some will be absorbed by other elements. Therefore, an efficiency factor must be considered in the preceding equation, η_{EVAP}:

$$H_{EVAP} = \eta_{WATER} \times V_{WATER} \times KA_{WATER}$$

Equation 24

The heat dissipated by evaporation will decrease the global cell temperature by an amount ΔT_{EVAP}:

$$\Delta T_{EVAP} = H_{EVAP} \times KA_{AIR}$$

Equation 25

Equation 11 must now be updated to include this term:

$$T_{CELL}^{k+1} = T_{CELL}^{k} + \Delta T_{CELL}(\Delta t) - \Delta T_{EVAP}$$

Equation 26

When the cell temperature drops below the fire point (T_{MAX}), the combustion stops and all the water thrown over the cell will be used to dissipate heat using the following rules:

$\Delta T_{EVAP} = (T_{CELL} - T_{BOIL})\ K_{WATER}$, if $T_{CELL} > T_{BOIL}$

$\Delta T_{EVAP} = 0$, if $T_{CELL} \leq T_{BOIL}$

When the cell temperature drops below the fire point (T_{MAX}), the combustion stops and all the water thrown over the cell will be used to dissipate heat using the following rules:

$\Delta T_{EVAP} = (T_{CELL} - T_{BOIL})\ K_{WATER}$, if $T_{CELL} > T_{BOIL}$

$\Delta T_{EVAP} = 0$, if $T_{CELL} \leq T_{BOIL}$

A.4. Modeling Agent Mobility In Pyrosim

In order to move within the Pyrosim environment, agents may control two basic variables: acceleration intensity and relative direction. Both acceleration and direction are defined in relation to the tangential plane (figure 1).

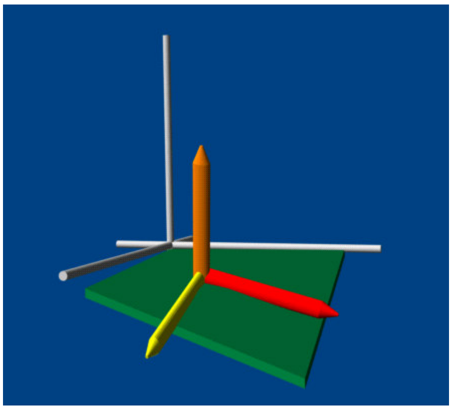

Figure A. 13. A Pyrosim cell. The Agent is located at the origin of the coloured axis

We will now try to define the equations governing the movement of Agents in Pyrosim.

Let us then consider an Agent located at position $P(x_i,y_i,z_i)$. Note that z, terrain height, is a function of both x,y:

$$z = T(x, y)$$

Equation 27

We can determine the tangential plane to the terrain in point $P(x_i,y_i,z_i)$ by first calculating the partial derivatives of $T(x,y)$:

$$\left.\frac{\partial z}{\partial x}\right|_{\substack{x=x_i\\y=y_i}} = \left.\frac{\partial T(x,y)}{\partial x}\right|_{\substack{x=x_i\\y=y_i}} = z_x'(x_i, y_i)$$

Equation 28

and:

$$\left.\frac{\partial z}{\partial y}\right|_{\substack{x=x_i\\y=y_i}} = \left.\frac{\partial T(x,y)}{\partial y}\right|_{\substack{x=x_i\\y=y_i}} = z_y'(x_i, y_i)$$

Equation 29

These values represent the inclination of the tangent plane to point $P(x_i,y_i,z_i)$, regarding the X and Y axis respectively. From these inclination values we can determine the angles θ_x and θ_y that the tangent vectors $\vec{t}_x$ and $\vec{t}_y$ establish with the base vectors $\vec{u}_x$ and $\vec{u}_y$ respectively:

$$\theta_x = arctg(z_x'(x_i, y_i))$$

Equation 30

and:

$$\theta_y = arctg(z_y'(x_i, y_i))$$

Equation 31

In figure 1 we can depict vector $\vec{t}_x$, the red one, and $\vec{t}_y$, the yellow one. We can now define the tangent vectors $\vec{t}_x$ and $\vec{t}_y$ using the base vectors $\vec{u}_x$, $\vec{u}_y$ and $\vec{u}_z$:

$$\vec{t}_x = \cos(\theta_x)\cdot\vec{u}_x - \sin(\theta_x)\cdot\vec{u}_z$$

Equation 32

$$\vec{t}_y = \cos(\theta_y)\cdot\vec{u}_y - \sin(\theta_y)\cdot\vec{u}_z$$

Equation 33

We can create local basis by including in the set of vectors $\{\vec{t}_x, \vec{t}_y\}$ the vector that is normal to the plane they define. The normal vector, $\vec{v}_z$ is calculated by:

$$\vec{v}_z = \vec{t}_x \otimes \vec{t}_y = \sin(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_x + \cos(\theta_x)\cdot\sin(\theta_y)\cdot\vec{u}_y + \cos(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_z$$

Equation 34

The basis vector, $\vec{t}_z$, is obtained by normalizing $\vec{v}_z$:

$$\vec{t}_z = \frac{\vec{v}_z}{\left\|\vec{v}_z\right\|} = \frac{\sin(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_x + \cos(\theta_x)\cdot\sin(\theta_y)\cdot\vec{u}_y + \cos(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_z}{\sqrt{\sin^2(\theta_x)\cdot\cos^2(\theta_y) + \cos^2(\theta_x)\cdot\sin^2(\theta_y) + \cos^2(\theta_x)\cdot\cos^2(\theta_y)}}$$

Equation 35

$\vec{t}_z$ may be simplified:

$$\vec{t}_z = \frac{\sin(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_x + \cos(\theta_x)\cdot\sin(\theta_y)\cdot\vec{u}_y + \cos(\theta_x)\cdot\cos(\theta_y)\cdot\vec{u}_z}{B}$$

Equation 36

With

$$B = \sqrt{\cos^2(\theta_y) + \cos^2(\theta_x)\cdot\sin^2(\theta_y)}$$

Equation 37

or

$$B = \sqrt{\cos^2(\theta_x) + \cos^2(\theta_y)\cdot\sin^2(\theta_x)}$$

Equation 38

We now have local tangential basis, $[T]$, given by:

$$[T] = [M]\cdot[U]$$

$$\begin{bmatrix} \vec{t}_x \\ \vec{t}_y \\ \vec{t}_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_x) & 0 & -\sin(\theta_x) \\ 0 & \cos(\theta_y) & -\sin(\theta_y) \\ \frac{\sin(\theta_x)\cdot\cos(\theta_y)}{B} & \frac{\cos(\theta_x)\cdot\sin(\theta_y)}{B} & \frac{\cos(\theta_x)\cdot\cos(\theta_y)}{B} \end{bmatrix} \cdot \begin{bmatrix} \vec{u}_x \\ \vec{u}_y \\ \vec{u}_z \end{bmatrix}$$

Equation 39

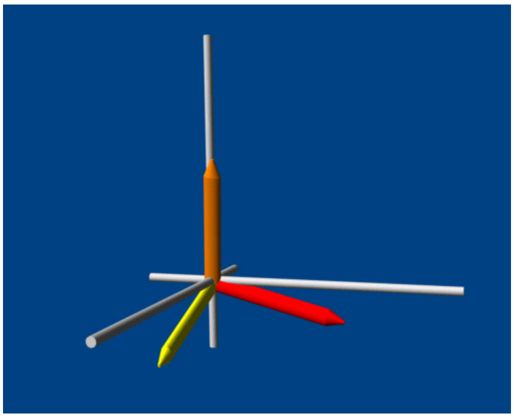

Figure A. 14. The tangential basis.

Alternatively, we can express the global basis vectors, $[U]$, as a linear combination of the local basis, $[T]$, by inverting $[M]$:

$$[U] = [M]^{-1} \cdot [T]$$

$$\begin{bmatrix} \vec{u}_x \\ \vec{u}_y \\ \vec{u}_z \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta_x)}{B^2} & \frac{-\cos(\theta_x)\cdot\sin(\theta_x)\cdot\sin(\theta_y)}{B^2} & \frac{\sin(\theta_x)\cdot\cos(\theta_y)}{B} \\ \frac{-\sin(\theta_x)\cdot\sin(\theta_y)\cdot\cos(\theta_y)}{B^2} & \frac{\cos(\theta_y)}{B^2} & \frac{\cos(\theta_x)\cdot\sin(\theta_y)}{B} \\ \frac{-\sin(\theta_x)\cdot\cos^2(\theta_y)}{B^2} & \frac{-\cos^2(\theta_x)\cdot\sin(\theta_y)}{B^2} & \frac{\cos(\theta_x)\cdot\cos(\theta_y)}{B} \end{bmatrix} \cdot \begin{bmatrix} \vec{t}_x \\ \vec{t}_y \\ \vec{t}_z \end{bmatrix}$$

Equation 40

Now that we have defined the appropriate geometrical relations, we can start deriving the equations that govern the movement of the Agent in the tangential plane. As mentioned previously, the Agent controls its movement by manipulating its acceleration intensity, A_{AGENT}, and direction ϕ_{AGENT}. Acceleration vector in the tangential plane is given by:

$$\vec{a}_{AGENT} = A_{AGENT} \cdot \cos(\phi_{AGENT}) \cdot \vec{t}_x + A_{AGENT} \cdot \sin(\phi_{AGENT}) \cdot \vec{t}_y$$

Equation 41

We must now include the effect of gravity in the Agent acceleration. We will first express gravitational force in the tangential basis:

$$\vec{g} = -g \cdot \vec{u}_z = \frac{g \cdot \sin(\theta_x) \cdot \cos^2(\theta_y)}{B^2} \cdot \vec{t}_x + \frac{g \cdot \cos^2(\theta_x) \cdot \sin(\theta_y)}{B^2} \cdot \vec{t}_y - \frac{g \cdot \cos(\theta_x) \cdot \cos(\theta_y)}{B} \cdot \vec{t}_z$$

Equation 42

Adding the appropriate components:

$$\vec{a}_{tx} = \left(A_{AGENT} \cdot \cos(\phi_{AGENT}) + \frac{g \cdot \sin(\theta_x) \cdot \cos^2(\theta_y)}{B^2} \right) \cdot \vec{t}_x$$

Equation 43

$$\vec{a}_{ty} = \left(A_{AGENT} \cdot \sin(\phi_{AGENT}) + \frac{g \cdot \cos^2(\theta_x) \cdot \sin(\theta_y)}{B^2} \right) \cdot \vec{t}_y$$

Equation 44

$$\vec{a}_{tz} = 0 \cdot \vec{t}_z$$

Equation 45

Agent controlled acceleration intensity is limited to $\frac{\sqrt{2}}{2} G = 0.707 \times 9.8$ m/s, which corresponds to a 45 degrees slope limit. Therefore, Agents will not be able to climb hills with slopes higher than 45 degrees, unless they accelerate sufficiently before starting to climb. Additionally, they will not be able to stop their movement when descending hills with slopes higher than 45 degrees, which they should avoid.

We will now derive Agent velocity and position equations. Derivation will be made for the $\vec{t}_x$ axis but the procedure for the $\vec{t}_y$ axis is similar.

$$\frac{\partial v_{tx}}{\partial t} = a_{tx}$$

$$v_{tx}(t) = v_{tx}(0) + a_{tx} \cdot t$$

$$v_{tx}(t+\Delta t) - v_{tx}(t) = a_{tx} \cdot \Delta t$$

Equation 46

Based on the last equation, the Pyrosim will update Agent velocity using the following approximations:

$$v_{tx}[k] = v_{tx}[k-1] + a_{tx}[k] \cdot \Delta t[k]$$

Equation 47

where $a_x[k]$ is the acceleration given by equation BBB. Velocity is limited to 8 m/s, which corresponds to running forward, and –2 m/s, which corresponds to moving backwards quickly.

$$\frac{\partial^2 x}{\partial t^2} = a_x$$

$$x_t(t) = x_t(0) + v_{tx}(0) \cdot t + \frac{1}{2} a_{tx} \cdot t^2$$

Equation 48

Then:

$$\begin{aligned}
&\Delta x_t = x_t(t+\Delta t) - x_t(t) = v_{tx}(0) \cdot \Delta t + \frac{1}{2} a_{tx} \cdot (t+\Delta t)^2 = \\
&v_{tx}(0) \cdot \Delta t + a_{tx} \cdot t \cdot \Delta t + \frac{1}{2} a_{tx} \cdot (\Delta t)^2 = \\
&(v_{tx}(0) + a_{tx} \cdot t) \cdot \Delta t + \frac{1}{2} a_{tx} \cdot (\Delta t)^2 = \\
&v_{tx}(t) \cdot \Delta t + \frac{1}{2} a_{tx} \cdot (\Delta t)^2
\end{aligned}$$

Equation 49

Pyrosim will update Agent position using the equations:

$$\Delta x_{tx}[k] = v_{tx}[k] \cdot \Delta t + \frac{1}{2} a_{tx}[k] \cdot (\Delta t)^2$$

$$\Delta y_{ty}[k] = v_{ty}[k] \cdot \Delta t + \frac{1}{2} a_{ty}[k] \cdot (\Delta t)^2$$

Equation 50

We must express this difference back in the absolute basis $\left\{ \vec{u}_x, \vec{u}_y, \vec{u}_z \right\}$:

$$\Delta x[k] = \Delta x_{tx}[k] \cdot \cos(\phi)$$
$$\Delta y[k] = \Delta y_{ty}[k] \cdot \sin(\phi)$$

Equation 51

Finally, we may obtain:

$$x[k] = x[k-1] + \Delta x[k]$$
$$y[k] = y[k-1] + \Delta y[k]$$
$$z[k] = T(x[k], y[k])$$

Equation 52