An Action Selection Architecture for Autonomous Virtual Humans in Persistent Worlds

Etienne de Sevin

AN ACTION SELECTION ARCHITECTURE FOR AUTONOMOUS VIRTUAL HUMANS IN PERSISTENT WORLDS

THÈSE N^{O} 3468 (2006)

PRÉSENTÉE LE 24 MARS 2006

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Laboratoire de réalité virtuelle

SECTION D'INFORMATIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Etienne de SEVIN

DEA d'Intelligence artificielle, Université de Paris VI, France
et de nationalité française

acceptée sur proposition du jury:

Prof. E. Charbon, président du jury
Prof. D. Thalmann, directeur de thèse
Prof. L. Canamero, rapporteur
Dr J.-Y. Donnart, rapporteur
Prof. A. Ijspeert, rapporteur

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2006

Abstract

Nowadays, virtual humans such as non-player characters in computer games need to have a strong autonomy in order to live their own life in persistent virtual worlds. When designing autonomous virtual humans, the action selection problem needs to be considered, as it is responsible for decision making at each moment in time. Indeed action selection architectures for autonomous virtual humans need to be reactive, proactive, motivational, and emotional to obtain a high degree of autonomy and individuality. The thesis can be divided into three parts.

In the first part, we define each word of our title to precise their sense and raise the problematic of this work. We describe also inspirations from several domains that we used to design our model because this thesis is highly multi-disciplinary. Indeed, decision-making is essential for every autonomous entity and is studied in ethology, robotics, computer graphics, computer sciences, and cognitive sciences. However, we have chosen specific techniques to implement our model: hierarchical classifier systems and a free flow hierarchy.

The second part of this thesis describes in detail our model of action selection for autonomous virtual humans. We use overlapping hierarchical classifier systems, working in parallel, to generate coherent behavioral plans. They are associated with the functionalities of a free flow hierarchy for the spreading of activation to give reactivity and flexibility to the hierarchical system. Moreover several functionalities are added to enhance and facilitate the choice of the most appropriate action at every time according to the internal and external influences.

Finally, in the third part of this thesis, a complex simulated environment is created for testing the model and its functionalities with many conflicting motivations. Results demonstrate that the model is sufficiently efficient, robust and flexible for designing motivational autonomous virtual humans in persistent worlds. Moreover, we have just started to investigate on the emotional level which has to be improved in the future to have more subjective and adaptive behaviors and also manage social interactions with other virtual humans or users. Applied to video games, non player characters are more interesting and believable because they live their own life when people don't interact with them.

Keywords: Behavioral Animation, Real-time, Virtual Humans, Action selection, Motivations, Emotions, Autonomy, Individuation, Personality.

Résumé

De nos jours, les humains virtuels ont besoin d'une grande autonomie pour pouvoir vivre leur propre vie dans des mondes virtuels persistants comme les personnages non joueurs dans les jeux vidéo. Lors de la conception d'humains virtuels autonomes, la problématique de la sélection de l'action doit être prise en compte car elle est responsable de leur prise de décision à chaque instant. En effet, les architectures de sélection de l'action pour les humains virtuels autonomes doivent être réactives, dirigées par des buts, et intégrer des motivations et des émotions pour obtenir un haut niveau d'autonomie et d'individualité. Cette thèse peut être divisée en trois parties.

Dans la première partie, nous définissons chaque mot de notre titre pour en préciser leur sens et poser la problématique de ce travail. Nous décrivons ensuite les domaines dont nous nous sommes inspirés pour élaborer notre modèle car le sujet de ce travail est très multidisciplinaire. En effet, la prise de décision est essentielle pour toute entité autonome et est étudiée en éthologie, robotique, infographie, informatique, et dans les sciences cognitives. Cependant nous avons choisi certaines techniques spécifiques pour implémenter notre modèle parmi toutes celles possibles : les systèmes de classeurs hiérarchiques et les hiérarchies à libre flux.

La seconde partie de cette thèse décrit en détail notre modèle de sélection de l'action pour des humains virtuels. Nous avons utilisé des systèmes de classeurs hiérarchiques, fonctionnant en parallèle, pour générer des plans de comportements cohérents. Ils sont associés avec les fonctionnalités des hiérarchies à libre flux pour la propagation de l'activité car elles donnent une grande flexibilité et réactivité aux systèmes hiérarchiques. Plusieurs fonctionnalités ont été ajoutées pour améliorer et faciliter le choix de l'action la plus appropriée, à chaque instant, par rapport aux influences internes et externes.

Finalement, dans la troisième partie de la thèse, un environnement virtuel complexe est créé avec beaucoup de motivations conflictuelles pour tester notre architecture et ses fonctionnalités. Les résultats démontrent que le modèle est suffisamment efficace, robuste, et flexible pour concevoir des humains virtuels motivés et autonomes dans des mondes virtuels persistants. De plus, nous venons de commencer les investigations au niveau des émotions dans notre modèle et nous projetons de continuer dans le futur pour avoir des comportements plus subjectifs et plus adaptés aux différentes situations ainsi que pour gérer les interactions sociales avec d'autres humains virtuels ou des utilisateurs. Appliquée aux jeux vidéo, notre architecture de sélection de l'action pour des humains virtuels autonomes dans des mondes persistants rendrait les personnages non-joueurs plus intéressants et plus réalistes car ils pourraient vivre leur propre vie lorsqu'ils ne sont pas en interaction avec les joueurs.

Mots-clés : Animation comportementale, Temps réel, Humains virtuels, Selection de l'action, Motivations, Emotions, Autonomie, Individualisation, Personnalité.

Remerciements

Tout d'abord, je voudrais remercier mon directeur de thèse, le Pr. Daniel Thalmann, pour m'avoir permis de faire mon doctorat au VRLab dans les meilleures conditions pendant les nombreuses années passées au laboratoire.

Je tiens à exprimer toute ma reconnaissance envers Dr. Jean-Yves Donnart, Pr. Lola Cañamero et Pr. Auke Ijspeert d'avoir accepté d'être rapporteurs à ma thèse et envers le Prof. Edoardo Charbon, président de mon jury. Ce fut un plaisir de leur présenter ma thèse. J'ai beaucoup apprécié leurs judicieuses et intéressantes remarques sur mon travail. Celles-ci m'ont permis de prendre du recul par rapport ma thèse.

Je voudrais aussi remercier toutes les personnes qui m'ont aidé dans cet exigeant travail. Stéphanie et Mirouille pour la beauté du graphisme et de l'animation de ma simulation, Barbara pour ses inlassables corrections de mon anglais, Sébastien pour la résolution des problèmes de VHD++ grâce à sa grande expérience de la programmation, Renaut pour m'avoir repris l'administration Windows au bon moment, à Julien pour ses bons conseils en ce qui concerne la soutenace et Josiane pour son aide précieuse et son soutien.

J'ai aussi beaucoup apprécié l'ambiance au VRLab. Merci a Pascal pour son patriotisme fribourgeois, ses judicieux conseils, les sorties à ski et les soirées des grands ducs avec Anthony et Béa (grands surfeurs et golfeurs) ; à Helena pour son apport en musique, qui m'a bien aidé durant l'écriture de ma thèse ; à Bruno pour les longues discussions philosophiques sur des sujets très variés ; à Benoît pour l'ambiance délurée en soirée ; à Sylvain pour les bonnes parties de jeu vidéo passées ensemble ; à Rachel pour les bonnes petites soirées et les ballades, à Raquel pour mon entraînement sportif et les sorties insolites ; à Christelle et Anthony pour les soirées jeux ; à Pablo pour ses bras de fer et à tous les autres personnes avec qui j'ai passé un bon moment durant cette thèse.

Enfin, je remercie mes proches : mes amis et ma famille. Merci à Dorothée et à Marilyne pour leur soutien et leurs encouragements, à ma mère pour avoir eu toujours confiance en moi et pour son inconditionnel support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The aim of this thesis consists in designing an action selection model for autonomous virtual humans that makes their decisions continuously in real-time according to the internal and external factors in a bottom-up approach, i.e., increasing the complexity progressively. In the end, they can live their own lives in their persistent environment and have more interesting and believable behaviors. In the following sections, the words of our main title are defined in order to precise their sense and also to illustrate the context, the main objective, the approach, and the main constraint of this thesis. Finally, the organization of the thesis is presented.

1.1 Context - virtual humans

Virtual humans can be defined as software entities that look and act like people but live in simulated environments. With the untidy problems of sensing and acting in the physical world thus dispensed, the focus of virtual human research is on capturing the richness and dynamics of human behavior. The potential applications of this technology are considerable. History students could visit ancient Greece and debate Aristotle. Patients with social phobias could rehearse threatening social situations in the safety of a virtual environment. A variety of applications are already in progress, including education and training, therapy, marketing, and entertainment [Gratch 02]. With the new developments of digital and interactive television and multimedia products, there is a need for systems that provide designers with the ability to embed real-time simulated humans in games, multimedia titles and film animations [Magnenat-Thalmann 04]. Building a virtual human is a complex, time consuming and multidisciplinary effort, joining several Computer Science areas: artificial intelligence, computer graphics, geometric modeling and multimodal interfaces with a range of issues from ethology to social science. Virtual humans must act and react in their simulated environment, drawing on the disciplines of automated reasoning and planning, providing human bodies that can be controlled in real time delves into computer graphics and animation. As an agent looks like a human, people expect it to behave like one as well and will be disturbed by, or misinterpret, discrepancies from human norms. Thus, virtual human research must draw heavily on psychology and communication theory to appropriately convey nonverbal behavior, emotion, and personality. Although graphics technology allows the creation of games in which the environment looks incredibly realistic, the behavior of computer controlled characters (referred to as Non Player Characters) often leads to a shallow, and unfulfilling game experience [Namee 01]. Badler [Badler 99] explains well the difficulty of designing virtual humans in a general way and not just for an application as it is often done in video games: "Why are real-time virtual humans so difficult to construct? After all, anyone who can watch a movie can see marvelous synthetic animals, characters, and people. But they are typically created for a single scene or movie and are neither autonomous nor meant to engage in interactive communication with real people. What makes a virtual human is not just a well-executed exterior design, but movements, reactions, self-motivated decision making, and interactions that appear "natural," appropriate, and contextually sensitive". Figure 1.1 summarizes the requirements for designing virtual humans in different applications.

Application	Appearance	Function	Time	Autonomy	Individuality
Cartoons	high	low	high	low	high
Games	high	low	low	medium	medium
Special Effects	high	low	high	low	medium
Medicine	high	high	medium	medium	medium
Ergonomics	medium	high	medium	medium	low
Education	medium	low	low	medium	medium
Tutoring	medium	low	medium	high	low
Military	medium	medium	low	medium	low

Figure 1.1: Requirements of representative virtual human applications

One of the main goals for the research in virtual humans is to design virtual humans that can be used for all applications with believable and coherent behaviors. According to Thalmann [Magnenat-Thalmann 04], "The ultimate research objective is the simulation of Virtual Worlds inhabited by a Virtual Human Society, where Virtual Humans will co-operate, negotiate, make friends, communicate, group and break up, depending on their likes, moods, emotions, goals, fears, etc. But such interaction and corresponding groups should not be preprogrammed. Behavior should emerge as a result of a multi-agent system sharing a common environment, in our case, sharing a Virtual Environment. If we model the distinctive entity, there will be groups of different behaviors (not programmed explicitly) as a result of the interaction of common distinctive behaviors... Behavior models should be developed that are simple enough to allow for real-time execution of group of agents, yet still sufficiently complex to provide interesting behaviors". The notion of individuality is very important to obtain interesting and believable autonomous virtual humans interacting in a society and having their own goals, moods, emotions, personalities... In this case, each virtual human reacts by his own and differently according to the others and his internal and external states. External states correspond to the changes of their environment, to the other agents and also to the actions of real humans interacting with the virtual word.

1.2 Objective - autonomy

Autonomy means that it provides the ability to exercise choice, which is particularly relevant in context of goals and goal-directed behavior, as in Calstelfranchi's notions of goal (or motivational) autonomy [Castelfranci 95]. Autonomy is recognizably and undeniably a critical issue in the field of intelligent agents and multi-agent systems, yet it is often ignored or simply assumed. Thus traditional AI systems and most robots are automatic but not autonomous; they are not independent of the control of their designers. Autonomous agents are able to generate their own goals, to select between multiple alternative goals to pursue, and to decide to adopt goals from others (to further their own ends) [Luck 03].

Autonomy is the key for believability of virtual humans. If they are autonomous, they can "live" their own life without external interventions and make their own choices of actions according to their internal and external factors. However to have an autonomy in a "strong sense" [Luck 98], the system should include motivations and the choice should be specific for each virtual human. This is the autonomy at the level of choices of behaviors. The virtual humans should also be autonomous at the level of behaviors, i.e. if the decision making of the virtual humans chooses to do a specific action; he should be able to do it no matter where he is. This ability implies chaining several other actions. For example, if the virtual human has chosen to accomplish the action "go to the kitchen to eat something", he will need to chain e.g. walk to the corridor, walk to the kitchen, take the food and put it in the mouth to finally achieve the chosen action. Moreover some of these sub-actions such as "walk to the corridor" imply other sequences of actions at each iteration like go right, go straight... In the end, there are several level of autonomy to obtain autonomous virtual humans.

Concerning autonomy in behavioural choice, several levels exist depending on the importance of the user control on the virtual humans. Blumberg [Blumberg 96] defines tree levels of autonomy:

- The character is a direct extension of the user, but the desired level of interaction is such that the user wishes to provide control at a high level and rely on the competence of the character to accomplish the task. The classic example of this is a web-based avatar which knows how to move around in a space without going through walls, and perhaps who "knows" the basics of interpersonal skills (e.g. knows how to face another avatar when "talking to it").

- The character is not directly driven by the user but interacts with him and other characters in a relatively structured environment. A non-player character in a multiplayer game, or an adaptive opponent in a single player game, or a companion in an interactive story-telling environment illustrate this level of autonomy. More autonomy is required than in the first case because it may have non-trivial choices to make in real-time, and these choices may depend on the character's goals and motivations as well as on its past history.

- The character is intended to give the illusion of being "alive" and of having an existence independent of the user such as "digital pets". Here the interaction and environment are potentially much less structured than in the previous two cases, and the time-span of interaction is measured in weeks or months. Further, it is not sufficient to simply be capable of autonomous action. The character must possess "life-like" qualities as well.

Fundamentally, autonomy is about choices, and about being self-contained. The implicit assumption is that the agent is constantly faced with non-trivial choices, and must decide on its own how to respond. It is self-contained in the sense that it does not rely on an external entity, i.e., a human or a centralized decision-maker to make its decisions for it.

1.3 Problem - action selection

The problem of action selection is central each time autonomous entities such as robots, virtual characters, or humans are designed. The system should decide what to do next according to its internal and external information without outside interventions. Action selection is a control structure for an autonomous agent (see Figure 1.2) and can be considered as the mind of the agent. The continuing task of mind is to produce the agent's next action to answer the only really significant question there is: what shall I do next? Franklin [Franklin 95] argues the following seven positions which briefly sketch the Action Selection Paradigm of mind:

- The overriding task of Mind is to produce the next action.
- Mind is better viewed as a continuous, as opposed to a boolean notion.
- Mind is aggregate rather than monolithic.
- Mind is enabled by a multitude of disparate mechanisms.
- Mind operates on sensations to create information for its own use.
- Mind uses prior information to produce actions by a reconstructive process rather than by retrieval.
- Mind, to some degree, is implementable on machines.

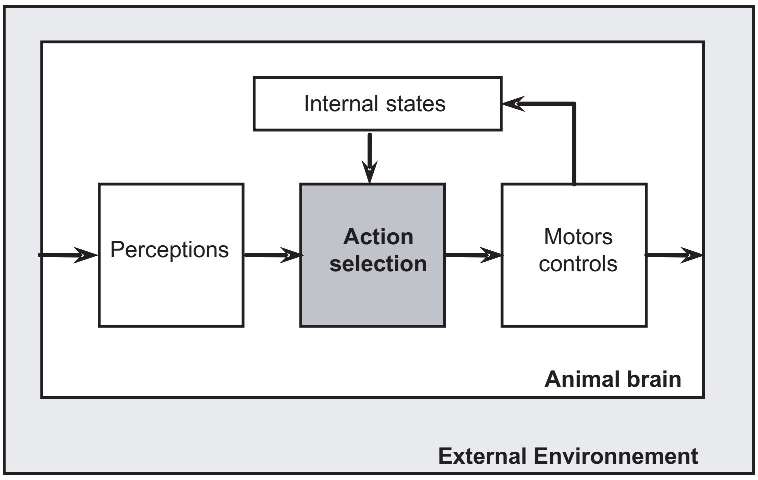

Figure 1.2: The action selection module inside the animal brain

However it is not easy to implement an agent mind. Most of the action selection models come from ethology observed on animals [Tinbergen 51, McFarland 75, Baerends 76, Dawkins 76, Lorenz 81, Toates 83]. Some researchers have implemented action selection mechanisms [Maes 90, Brooks 86, Rosenblatt 88, Tyrrell 93a, Blumberg 96, Bryson 00]. This problem is still studied in ethology and in psychology because it can be extended to humans. It can be resumed as "how to choose the appropriate actions at each point in time so as to work towards the satisfaction of the current goal (its most urgent need), paying attention at the same time to the demands and opportunities coming from the environment, and without neglecting, in the long term, the satisfaction of other active needs" [Cañamero 97]. Every autonomous agent must repeatedly solve the action selection problem to know what it will do next.

At every instant the agent should choose the actions which make the most sense, given its internal state (e.g. food and water needs), its perception of its environment, and its repertoire of possible actions. Moreover, the temporal pattern of its behavior should make sense as well. If it is working on a given goal, it should continue working on that goal until either the goal is satisfied or something better comes along. That is, it should be able to balance persistence with opportunism and have a sense of whether it is making progress, i.e., it should not get stuck in "mindless loops" [Maes 90]. Many problems are linked with action selection such as action persistence, evaluation of the action choice, chaining actions to obtain coherent behaviors, authorizing opportunist and compromise behaviors... [Blumberg 94]

1.4 Constraint - persistent worlds

A persistent world is a virtual world (often for online video games) which is always available and world events happen continually. In persistent worlds, virtual humans can not be controlled all the time by designers or users. They have to continue to take decisions even if the users have left or their tasks are finished. They should be really autonomous. Moreover they should be believable in the sense that they should behave like humans and live their lives by their own. They should be able to make decisions continuously according to their perceptions of the virtual environment (and its changes), their need, their goals and the emotions. For example, non-player characters in role-play or adventure video games should act and react with as much believability as possible. However, these non-player characters simply wait for the player to arrive at their location and then, either await the player's involving them in some interaction, or play through a scripted sequence [Nareyek 01]. This kind of character could be considered static agent (see Figure 1.3). The persistence of these agents refers to the fact that they are always modeled, at least to some extent, regardless of the player's location in the game world. If they have a strong autonomy - irrespective of the player's location - it would greatly add to the sense of immersion. They can make decisions according to their motivations and their perceptions. In this case, they live their own lives

Figure 1.3: Example of static non-player characters waiting for interactions in World of Warcraft game [World of Warcraft 05]

In real-time persistent environments, the virtual humans should be situated to react quickly to the changes around them. Situatedness denotes the predicament of being in a world. This implies that an agent has to obey the spatial and temporal conditions of the (simulated) world, but it also entails that it can exploit what the environment does for it [Rank 05]. In fact, signification comes from the environment whereas in symbolic (cognitive) architectures, it comes from the symbolic database. The characteristics of situated agents compared to symbolic agents are:

- Limitation of the knowledge (opposite in symbolic architectures)
- Necessary in dynamic or real-time environment
- Bottom-up approach versus top-down approach.
- Increasing the complexity progressively

- Beginning by implementing simple things and adding more and more things.
- The complexity of the architecture should emerge by the interactions such as intelligence or consciousness.

In situated systems, agents and the environment constitute complementary parts of a world where they can mutually affect each other. Situatedness places an agent in a context in which it is able to perceive its environment and in which it can (inter)act.

1.5 Organization of the thesis

This thesis begins with resuming knowledge from many domains that inspired me during the design of our action selection architecture for autonomous virtual humans in persistent worlds (Chapter 2). Indeed the subject of this thesis is very multi-disciplinary including ethology, robotics, computer sciences, computer graphics, and cognitive sciences. This is not an exhaustive review but just some notions that helps me in my work.

Chapter 3 presents the techniques used for implemented the model. We use reactive and goal-oriented Hierarchical classifiers systems [Donnart 94] associated with the functionalities of a free flow hierarchy [Tyrrell 93a] for the spreading of activition. Hierarchical classifier systems allow to have a coherent and robust behavior by finding sequence of actions into plans in order to achieve goals. Free flow hierarchy brings reactivity and flexibility to the hierarchical system necessary to effective action selection mechanisms.

Chapter 4 describes our action selection architecture for autonomous virtual humans in persistent worlds with its four levels and all the functionalities needed to choose the more appropriate actions at every moment in time. Chapter 5 and Chapter 6 detail its implementation and its test simulations. In VHD++ [Ponder 03] developed in the VRLab, a 3D virtual environment: an apartment is created in order to test the model and its functionalities. Chapter 7 shows the results that the model generates coherent and flexible behaviors over 65000 iterations or one hour and half simulation.

Finally Chapter 8 concludes that our model of action selection is enough robust and reactive for designing autonomous virtual humans in real-time in persistent worlds so that they can live their own lives according to internal and external factors. It presents also the limits and the possible future works

Chapter 2

Inspirations

2.1 Ethology

Ethology is the study of animal behaviour in situ and ethological perspective accepts that behaviours are the product of continuous agent-environment interactions and profitably incorporated into a computational architecture for autonomous animated creatures. It also concerns the animal origin of human beings and natural selection [Darwin 59].

According to Blumberg [Blumberg 96], ethology is a valuable source of ideas for three reasons.

- Ethologists address relevant problems, namely "how are animals so organized that they are motivated to do what they ought to do at a particular time" [McFarland 89]. Thus, they wrestle with the issues of relevance, coherence and adaptation within the context of animal behavior.

- Ethologists have a bias toward simple non-cognitive explanations for behavior. They stress that seemingly intelligent behavior can be the result of very simple rules or from the interaction of what Minsky later called "many little parts, each mindless by itself" [Minsky 85]. In addition they emphasize that the behavioral and perceptual mechanisms of animals have evolved to opportunistically take advantage of whatever regularities and constraints are afforded them by the environment.

- Ethologists tend to analyze behavior at the same level which we wish to synthesize it, i.e. in terms of black boxes such as "avoid" or "chew". Thus, they are less concerned with how these behaviors are implemented at the neural level, than with understanding what the behaviors are, and how they interact.

Ethology studies are very useful for modeling basic motivations or drives such as hunger, thirst, fatigue... Most of the action selection models come from ethological studies [Tinbergen 50, Baerends 76, McFarland 75, Dawkins 76, Lorenz 81, Toates 83]. Seeing the limitations of the traditional artificial intelligence techniques [Maes 93], many researchers have been interested by the ethology studies for implementing behavioral models in a "bottom-up" approach such as [Brooks 91, Tu 99, Maes 91, Tyrrell 94, Blumberg 94, Bryson 00, Guillot 00, Cañamero 97]...

2.1.1 Bottom-up approach

Indeed beginning to design directly decision architectures for virtual humans with all their complexity is difficult (done by the classical artificial intelligence). The "bottom-up" approach, the "nouvelle AI" approach [Brooks 91] and Minsky's theory of intelligence [Minsky 85] view intelligence as emerging from the interactions of "systems" inside one agent connected in specific ways on the contrary of "cognitive" models such as SOAR [Laird 87]. While designing first decision architectures with basic motivations generating simple goals is much easier. In this case, the decision architecture manages a level of complexity comparable with that of simple animals. Next, following the evolution of natural selection and progressively increasing the complexity of the architecture and the environment, behaviors become richer and closer to those of the humans. This approach is called "bottom-up" approach contrary to the traditional artificial intelligence considered as a "top-down" approach. The agents are situated and take the information directly from the environment. It can quickly adapt its behavior to the changes.

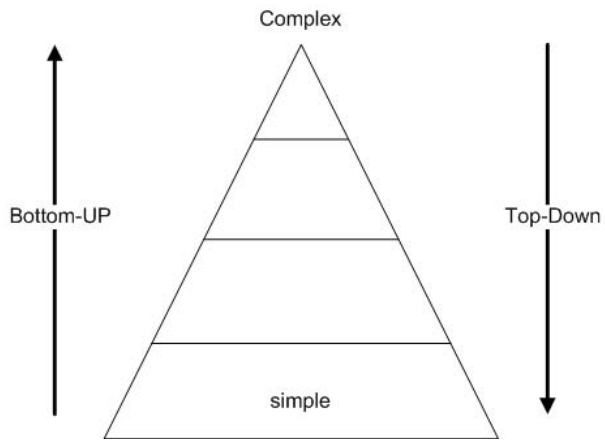

Figure 2.1: Top-down and bottom-up approaches

Whereas "bottom-up" approach allows reactivity and adaptability in real time, it can not manage complex behaviors such as virtual human behaviors. These behaviors need sequence of ordering actions with predefined knowledge inserted by the designers. For example, they design a kitchen where virtual humans "know" they can eat. For these behaviors, a "Top-down" approach is necessary with a behavioral planner. The ultimate goal is to try to understand the complexity of the human behaviors mixing the advantages of the "bottom-up" and "top-down" approaches.

2.1.2 Action selection

Action selection (defined in Section 1.3) is one of the main problems in ethology. It is still studied and can be resumed as what to do next.

Some requirements have to be respected to design an effective action selection mechanism [Tyrrell 93a, Seth 98]:

- Prioritise behaviour according to current needs, e.g., head for food if hungry, but don't fall into traps on the way.
- Allow contiguous action sequences to be strung together.
- Exhibit opportunism, for example by diverting to a nearby food source even if water is needed more.
- Balance dithering and persistence, e.g., by drinking until full and then eating until full instead of oscillating between food and drink.
- Interrupt current behaviour, for example, if a trap suddenly appears, the animal should change its course to avoid it.
- Deal with all subproblems, the action selection mechanism should cope effectively in all situations.
- Prefer consummatory over appetitive actions.
- Use all available information, both internal and external.
- Support real-valued sensors and produce directly useable outputs.
- Be extensible and support learning.
- Allow 'parallel actions' to be executed, for example, walking and talking.

From an evolutionary point of view and a "bottom-up" approach, motivations should be designed before emotions because emotions are specific to advanced animals and humans. The motivational brain systems are evolutionarily older than emotional and cognitive ones. The affect system provides the primary blueprints for cognition and decision making, as well as for action [Blumberg 96]. Action selection mechanisms should be first designed to satisfy motivations such as hunger, thirst, fatigue, etc. according to the external factors and then consider emotions. Moreover action selection is linked with the problem of attention. That is, how does the creature decide which features of its environment to attend to in order to assess the relative relevance of its behaviors ? [Bryson 00].

2.1.3 Persistence and compromises

The relevance (i.e. "do the right things") and the coherence (i.e. "show the right amount of persistence") of the behaviors are two important problems in ethology. How does the agent decide how long to persist in a given course of action, and how does it insure that its pattern of behavior over time is coherent (i.e. that competing behaviors don't work at cross-purposes)? [Blumberg 94]. An agent has only limited resources to satisfy its goals (e.g. it can only walk in one direction at a time). Thus, there needs to be some mechanism to arbitrate among the competing behaviors. Moreover, once a creature is committed to satisfy a goal, it makes sense for it to continue pursuing that goal unless something significantly more important comes along. This need for coherence of action and goal introduces two sub-problems [Humphrys 96]:

- **Temporal control** To provide just the right amount of persistence. Too little persistence and the creature will dither between behaviors that address competing goals. Too much persistence and the creature will not take advantage of unexpected opportunities. Or it may mindlessly pursue a given goal to the detriment of other goals.

- **Concurrency and compromise** Pursuit of a single goal should not preclude the expression of compatible behaviors which may address other goals. The system must be able to support concurrency, blending and modulation of motor actions in such a way that the creature still behaves coherently.

These two problems should be managed by action selection mechanisms in order to have a coherent and robust decision-making.

2.2 Robotics

In robotics, the environment is real, unpredictable and dynamic. Thus, symbolic artificial intelligence can not really deal efficiently with these types of environment due to its lack of interactions with the environment [Brooks 86]. As mentioned above, many researchers in robotics such as Brooks [Brooks 91], McFarland [McFarland 91], Meyer [Meyer 95], etc. are inspired by ethology and the "bottom-up" approach. They try to design robotic architectures that are more reactive and adaptive such as the Animat approach.

2.2.1 Animat approach

The objective of the animat approach [Meyer 95] is to contribute to the advancement of cognitive sciences through the study of how human intelligence is rooted in simpler adaptive behaviors inherited from animals, in a bottom-up, evolutionary and situated perspective. To fulfill its objectives, simulated systems or robots are conceived within more or less changing and unpredictable environments in order to exhibit autonomously adaptive capacities.

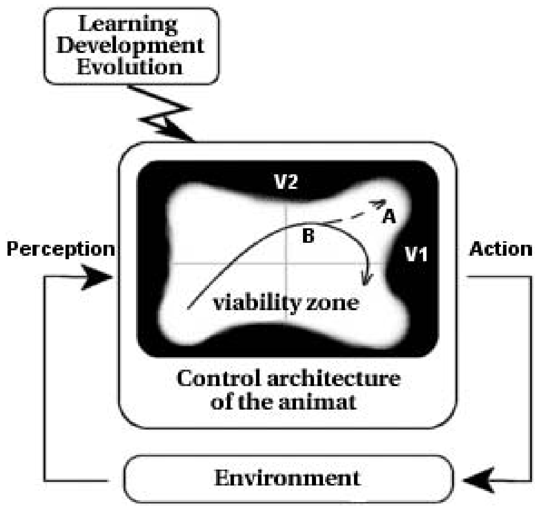

Figure 2.2: The Animat approach [Donnart 98]

The animat approach postulates that the mechanisms underlying adaptive behaviors must be studied and conceived in situation, i.e., by taking into account both the goals the animat tries to reach and the opportunities that the environment provides with respect to these goals with a minimum of human interventions. This approach is complementary to the one of symbolic artificial intelligence (see Section 2.1.1), which provides better performance when the environment characteristics can be predicted. At the interface of neurosciences, cognitive science, ethology and ecology, on the one hand, of computer science and robotics, on the other hand, the animat approach is a highly interdisciplinary research [Meyer 96, Guillot 00].

2.2.2 Robotic models

Brooks [Brooks 91] was the first, in the robotic field, to break the dogma of designing intelligence without representations because robots need quickly to take decisions in real-time and in the real word. He inspired many works, in particular Tyrrell's and Maes' works [Maes 93, Tyrrell 93a]. After the rupture of symbolic artificial intelligence, many models, called behavior-based models, were designed centered on behaviors and not on intelligence. However these models are limited in the complexity of the tasks. They can't well manage sequencing of ordering actions necessary for complex behaviors whereas hierarchical models of ethology can. Nowadays, multi-layered architectures are developed because they mix the advantages of behavior-based models and hierarchical organizations. In this case, decision architectures will be able to be reactive with opportunist behaviors but also to plan some sequences of ordering actions to fulfill specific complex goals.

2.2.3 Hierarchical classifier systems

A hierarchical classifier system [Donnart 94] (see Chapter 3.1 for more details) can generate reactive as well as goal-oriented behaviors because two types of rules exist in the rule base: external, to send actions directly to the motors, and internal, to modify the internal state of the classifier system. The message list contains only internal messages, creating the internal state of the system. The internal rules are used to plan sequences of actions and correspond to the internal state of the hierarchical classifier systems. They provide an internal context for the activation of the rules. The number of matching rules is therefore reduced, as only two conditions need to be fulfilled to activate rules in the base: the environmental information and the internal context of the system. Internal messages can be stacked in the message list until they have been carried out by specific actions. Depending on the situations, the architecture can react quickly to the changes of the environment or construct plans to fulfill some more complex goals. For example, an agent can move to a specific location from anywhere with the aim of preparing the agent to perform motivated actions that will satisfy his motivations.

2.3 Computer graphics

Computer sciences are useful for testing autonomous agent architectures. In robotics, researchers begin to simulate their robots to do the groundwork on the studied problem and then test it in reality with many other difficulties. Indeed simulated environments are much simpler than the real ones. While robotics is still dealing with the problem of autonomous path-planning in complex environment such as reaching a location and avoiding obstacles, computer science researchers try to implement complex behavior models because of the absence of reality constraints. It is easy to get round some difficulties in computer graphics. For example, modelisation of humans is very frequent and well advanced in computer graphics unlike in robotics. Everything is theoretically possible in simulation; however it is particularly difficult to design the complexity of human beings. Many architectures, such as SOAR [Laird 87], based on psychology theories, have tried to represent human reasoning for virtual humans. Computer graphics are a good test-bed for all the new theory of mind, consciousness, intelligence, etc. using virtual humans. In the end, it is easier to develop behavior models for virtual humans in computer graphics.

2.3.1 Virtual human realism

Figure 2.3: Graphic realism (from the final fantasy film made by Square)

With the new development of digital and interactive television and multimedia products, there is also a need for systems that provide designers with the ability to embed real-time simulated humans in games, multimedia titles, learning, training applications and film animations... [Magnenat-Thalmann 04]. For many years, 2D and 3D films have taken us along into virtual worlds that children but also adults could more and less believe in. For the last few years, graphical appearance has become more and more realistic due to the progress of computer graphics and the explosion of digital medias such as 3D animation films and video games (see Figure 2.3).

Nevertheless some graphical problems are still present. By using virtual humans, people feel more immersed because they can identify themselves with the virtual humans but they can also easily find defaults compared to the real humans. Thus, designing virtual humans is difficult to be realistic and believable because it implies many problems at several levels ranging from geometry to behaviors such as body motion control, body and facial deformations, cloth simulation, interactions with 3-D objects, etc. These problems can be avoided by using a middleware 3D engines [Ponder 03] which provides all the components needed for designing virtual human behaviors (see Section 5.1).

2.3.2 Behaviors in persistent worlds

Figure 2.4: Virtual humans in World of Warcraft game

Behaviors can be defined as arbitrary complex action patterns made in response to external stimuli or inferred internal motivational states. Nowadays the effort is more focused on behavior animation than on graphics. Indeed more autonomy and interesting behaviors could mend the lack of realism and believability that the virtual humans suffer in persistent worlds as non player characters in multi-massive online video games [World of Warcraft 05, Everquest2 04, Camelot 02] (see Figure 2.4).

In persistent worlds, the decision architecture should permanently choose the next action, i.e. its work is not finished after solving a specific task. Most of the behavioral architectures have a task-dependent and/or a time limited autonomy (e.g. interacting with users or find something in the environment). In this case, the simulation is finished when the virtual character has fulfilled a certain task while in persistent worlds, the decisions have to be taken continuously and virtual humans need to have a strong autonomy [Luck 01]. At certain moment in time, nobody can control him because he is always present in the virtual environment. Therefore he has to be able to take his decision by his own and live his own life. Moreover, persistent virtual worlds imply real-time decisions and dynamic environments: the virtual character should react adequately and rapidly to the changes of the environment.

2.3.3 Non-player characters

In computer games, developers are increasingly relying on "game artificial intelligence", i.e., behaviors of synthetic entities, to distinguish their game from those of competitors [Woodcock 00]. As autonomous virtual humans are also becoming more common in commercial role-playing games as non player characters, game developers are particularly concerned with goal-oriented architectures. Therefore, game industries are closer than ever with the academic research on virtual humans [Badler 99, Gratch 04, Magnenat-Thalmann 04]. and non-player characters in computer games [Laird 02, Isla 02, Nareyek 04, Thorisson 04, Baillie-de Byl 04].

Although graphics technology allows the creation of environments looking incredibly realistic, the behavior of non-player characters often leads to a shallow and unfulfilling game experiences [Namee 01]. Everyone who has played computer games has observed that characters controlled by the computer are neither very intelligent nor autonomous (even in The Sims [Sims 2 04]). For example, in role-playing games [The Elder Scrolls 3 03, NeverWinter Night 04, World of Warcraft 05], the non-player characters inhabiting persistent virtual worlds should give the illusion of living their own lives instead of staying static or having limited or scripted behaviors. When you play with your favorite character and you walk across a town, if each non-player character lives his own life when you don't interact with them instead of staying still or have predefined behaviors, the game will be more interesting and believable and give finally a fulfilling experience.

The non-player characters should have a behavioral architecture to be more autonomous and believable. However most of these architectures are efficient but have a contextual autonomy in the sense that they are designed to solve specific complex tasks (cognitive architectures), to follow scripted scenarios (virtual storytelling), or to interact with other agents (BDI architectures). Autonomous virtual humans in persistent worlds need to go on taking decisions according to their internal and external factors once complex tasks, scripted scenarios, or social interactions are finished.

2.4 Computer sciences

The study and development of autonomous agents has been an active field of research for several years. The question "is this agent autonomous" has interested many researchers in many fields such as robotics, computer graphics, etc. and some requirements have been deduced in order to classify the agents. To be autonomous, agents have to possess a decision-making architecture to decide what it will do next without external intervention. Many architectures have been implemented but in the end, they should be reactive to fit to environmental changes and coherent to be able to have structured sequences of actions in order to reach a final goal.

2.4.1 Definition of autonomous agents

According to Franklin and Grasser [Franklin 97], autonomous agent "is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

An intelligent agent is a versatile and adaptive system that performs diverse behaviours in its efforts to achieve multiple goals in a dynamic, uncertain environment [Morignot 96]. Blumberg [Blumberg 96] adds some notions to define an autonomous animated creature, which is an animated object capable of goal-directed and time varying behavior. The opposite of the virtual human in Figure 2.5.

Each autonomous agent is situated in, and is a part on some environment. Each senses its environment and act autonomously upon it. No other entity is required to feed it input, or to interpret and use its output. Each acts in pursuit of its own agenda,

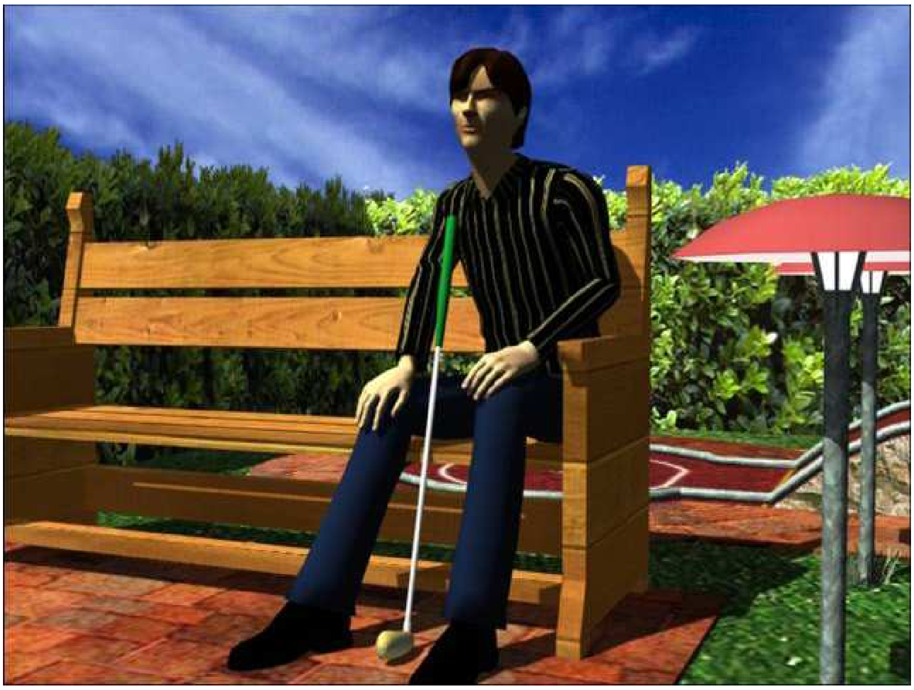

Figure 2.5: Example of non autonomous agent because he doesn't know what he will do next.

whether satisfying evolved drives as in humans and animals, or pursuing goals designed in by some other agent. Each acts so that its current actions may effect its later sensing, that is its actions effect its environment. Finally, each acts continually over some period of time [Franklin 97].

The notion of individuality is very important for autonomous agents because they should decide their actions according to internal and external states by their own. The final decision is made by the agents. Further, along with being reactive, an agent must also be proactive. That is, it must be able to take initiative and be opportunistic when necessary. The notion of planning their behaviors to anticipate the future actions is also necessary and to plan sequences of actions to reach a specific goal.

2.4.2 Requirements for designing autonomous agents

For designing complex autonomous agents that are situated within an environment, Wooldridge and Jennings [Wooldridge 95] define the following properties:

- reactivity and situatedness: agents perceive their environment and respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative, when appropriate; and learning from its own experience, its environment, and interactions with others.
- autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;

The notion of situatedness is often forgotten compared to the others and is very important in real-time environments. It implies to use the "bottom-up" approach (see Section 2.1.1). A situated agent is defined as an agent which [Steegmans 04]:

- is situated in an environment
- is driven by a survival/satisfaction function
- possesses resources of its own in terms of power and tools
- is capable of perceiving its environment (but to a limited extent)
- has practically no representation of its environment
- possesses skills
- can perhaps reproduce

Situatedness places an agent in a context in which it is able to perceive its environment and in which it can (inter)act. The agent also acts in such a way as to possibly influence what it senses at a later time. It is structurally coupled to its environment [Maturana 75, Maturana 80]. Situated agents do not use long-term planning to decide what action sequence should be executed, but select actions based on the locally perceived state of the world and limited internal state. Contrary to knowledge-based agents, situated agents do not emphasize internal modeling of the environment. Instead, they favor to employ the environment itself as a source of information. The environment can serve as a robust self-revising common memory for agents. This can unburden the distinctive agents from continuously keeping track of their knowledge about the system. The benefits of situatedness are well known:

- flexibility
- robustness
- efficiency

A well-known family of agent architectures for adaptive behavior are free-flow architectures [Tyrrell 93a].

2.4.3 Decision making architectures

For designing decision making architectures for virtual characters, two approaches exist. The "Top-Down" approach corresponds to traditional artificial intelligence including representations and the "Bottom-Up" approach which is often used in robotics including an ethnologist point of view and the concept of emergence. Then decision making architectures can be grouped into three broad categories: hierarchical ("top-down), behavior-based ("bottom-up") and three-layer (mixed). Nowadays the three-layered architectures are the most used by researchers.

Hierarchical architectures

The use of both hierarchical and fixed sequential orderings of behavior for action selection has been postulated since the time of the early ethologists [Lorenz 81]. It corresponds to the "top-down" approach. Hierarchical models mean multi-level systems where the modules are built into some kind of structure. Some modules have precedence over others, and control flows down to their submodules [Bryson 04] The advantages of such systems are clear: they reduce the combinatorial complexity of control. The set of actions which may be selected from is determined by context, including state internal to the agent. The choice of the most activated node is made at each level of the hierarchy neglecting the other possible choices. Early artificial intelligence research followed these hierarchical models, but ran into great difficulty in coping with dynamic environments in real time. While many problems seem to lend themselves to this kind of solution, there is normally a considerable burden of hand-design. Some authors also reject hierarchy in a controller altogether, on the basis that it results in bottlenecks, staged responses, and governed, unreactive behavior [Maes 91, Hendriks-Jansen 96]. Others design systems that exploit hierarchical order, but still maintain constant parallel processing [Tyrrell 93a, Blumberg 96].

Behavior-based architectures

Behavior-based architectures, used principally in robotics [Maes 93, Seth 98, Mataric 98, Arkin 98], follow the "Bottom-up" approach (see Section 2.1.1) and have been implemented to fix problems with traditional planning architectures:

- Constructing a complete plan before beginning action. A planner cannot determine whether a plan is viable before it is complete. Many plans are in fact formed backwards because of opportunities and changes in the environment.
- Taking too long to create a plan, thereby ignoring the demands of the moment.
- Being unable to create plans that contain elements other than primitive acts.
- Being unable to manipulate plans and goals.

Behavior-based models are used to implement fully reactive agent. A reactive system is designed from the beginning to be situated in a complex, dynamic environment, which it must constantly monitor and to which it must instantly react. They can respond quickly to new, unexpected or opportunistic situations in the environment whereas a traditional behavior planner will continue to execute its script until the end even if the intention of the agent or the conditions of the plans are changed. Reactive agents will notice and take decisions according to opportunities which can fulfill any of their goals. Moreover in reactive agents, the information is always up-to-date and consequently the behavior plan also. This is because no information is stored. All information is a reflection of the current environment.

Hybrid architectures

Although the fact that the majority of researchers accept that reactivity and modularity are good approaches for modeling cognition, architectures cannot be reduced to this [Bryson 04]. A hybrid approach has been established, where a behavior-based system is designed to work with a traditional AI planner. This approach is nowadays the most used by the researchers [Blumberg 95, Sloman 99, Donnart 96c, Bryson 00, Nareyek 01, Sevin 05]. It deduces the next action by searching a knowledge base for an act that will bring it closer to a goal. Traditionally, planners have micro-managed, scripting every distinctive motion. By making their elements semi-autonomous behaviors which will react or adapt to limited uncertainty, the planner themselves can be simplified.

The behavior-based plan execution was implemented bottom up to have as much useful capability as possible, where a useful capability is one which looked like it would simplify the design of the planner. Similarly, the planner was designed top down towards this interface, clarifying the nature of useful capabilities at which the behavior-based system should aim. This design method greatly reduced the complexity of the planner, increasing the complexity of the agent much less than this reduction, and thus reduced the overall system complexity. It also produced a robust system, capable of executing novel plans reliably despite... uncertainty [Malcolm 97].

2.5 Cognitive sciences

To model action selection mechanism for virtual humans, ethology is not enough and cognitive science contributions are necessary due to his multi-disciplinarily approach. Cognitive science is usually defined as the scientific study either of mind or of intelligence [Luger 94]. Practically every introduction to cognitive science also stresses that it is highly interdisciplinary; it is often said to consist of, take part in, and collaborate with psychology (especially cognitive psychology), artificial intelligence, linguistics and psycholinguistics, philosophy (especially philosophy of mind), neuroscience, logic, robotics, anthropology and biology (including biomechanics). Problems of motivations, emotions, mind... are studied from a global point of view

2.5.1 Motivations

The term motivation is used to describe "drives that constitute urges to action based on internal needs related with survival and self-sufficiency" [Cañamero 01]. Motivations can be seen as homeostatic processes which maintain a controlled physiological variable within a certain range. They involve arousal and satiation by specific type of stimulus, and vary as a function of deprivation [Cañamero 01]. The three mains functions of motivations [Kandel 95]:

- They steer behavior toward, or away from, a specific goal
- They increase general alertness and energize the individual to action
- They combine individual behavioral comportments into a goal-oriented sequence.

Motivations are a prerequisite for any cognitive system and are very relevant to emotions [Sloman 87], In action selection, motivational states guide the choice of the behaviors(s) that must be executed - those best contributing to the satisfaction of the most urgent current need(s). "They can be thought of as being concerned with appetitive processes that try to activate action as a response to deprivation" [Cañamero 01]. "Motivations have to be integrated in artificial systems to promote decision making, activity selection, and autonomy". Motivations coming from internal states of agents are often missing in computational agent-based systems [Luck 98]. "Autonomous" entities in the strong sense are goal-governed and self-motivated [Luck 03]. The self-generation of goals by the internal motivations is critical in achieving autonomy [Balkenius 93].

Ethology is a good based for modeling basic motivations coming from animals but, at a certain level of human complexity, the point of view of Psychology are necessary to design motivational decision model. Abraham Maslow [Maslow 54] is known for establishing the theory of a hierarchy of needs, writing that human beings are motivated by unsatisfied needs, and that some lower needs need to be satisfied before higher needs. According to Maslow, there are general types of needs (physiological, safety, love, and esteem) that must be satisfied before a person can act unselfishly. He called these needs "deficiency needs." As long as we are motivated to satisfy these cravings, we are moving towards growth, toward self-actualization.

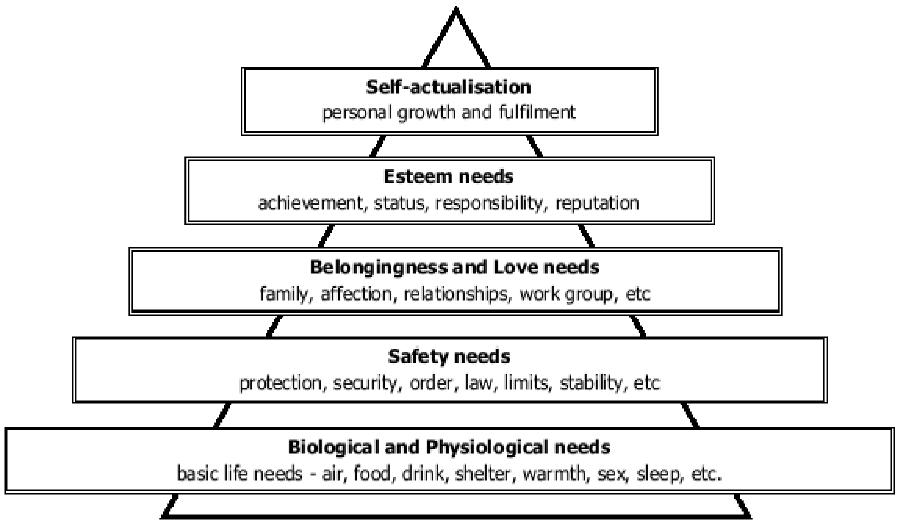

Figure 2.6: Maslow's hierarchy of needs

- **Physiological needs** are the very basic needs such as air, water, food, sleep, sex, etc. When these are not satisfied we may feel sickness, irritation, pain, discomfort, etc. These feelings motivate us to alleviate them as soon as possible to establish homeostasis. Once they are alleviated, we may think about other needs.

- **Safety needs** have to do with establishing stability and consistency in a chaotic world. These needs are mostly psychological in nature. We need the security of a home and family.

- **Love needs and belongingness** Humans have a desire to belong to groups: clubs, work groups, religious groups, family, gangs, etc. We need to feel that we are accepted and loved not sexually by other people.

- **Esteem needs** are divided into two types. First is self-esteem which results from competence or mastery of a task. Second, there is the attention and recognition that comes from others. This is similar to the belongingness level; however wanting to be admired has to do with the need for power.

- **Self- Actualization** is "the desire to become more and more what one is, to become everything that one is capable of becoming." People who have everything can maximize their potential. They can seek knowledge, peace, esthetic experiences, self-fulfillment, etc.

For designing virtual humans which are believable and realistic, this hierarchy of needs is a good source of inspiration and it can be implemented progressively following the complexity of the levels. The more layers you have implemented, the more complex and therefore realistic the behaviors of virtual humans are. To implement the first layer, i.e. physiological needs, ethology studies are very useful because physical needs have already well studied and many models have been realized [Toates 86, Maes 91, Tyrrell 94, Blumberg 94, Cañamero 97, Bryson 00]. For the second and the third layers, i.e. safety needs and love needs, theories and models of Emotions which are also well studied in psychology [Tomkins 84, Ortony 88, Frijda 95, LeDoux 96, Picard 97, Sloman 98, Rolls 99] and social interactions can be used. The fourth level, i.e. esteem needs, can be associated with the studies of consciousness, individuality and personality [Bogner 99, Franklin 01, Franklin 02, Bryson 03]. Finally the last level, i.e. self- actualization, can be considered as a consequence of the other levels and is complicated to implement. The modeling of mind (artificial mind) [Sloman 97] in computer sciences is a complex approach but has a great deal of future.

2.5.2 Emotions

"Motivations states are drives that constitute urges to action based on internal bodily needs related to self-sufficiently and survival, whereas emotions are the second-order modifiers or amplifiers of motivations and behaviors. Emotions enhance autonomy, adaptation, and social interactions in artificial and mixed-agent societies" [Cañamero 01].

Damasio [Damasio 94] defines an emotion as the combination of a mental evaluative process, simple and complex, with dispositional responses to that process, mostly toward the body proper, resulting in an emotional body state, but also toward the brain itself (neurotransmitter nuclei in brain stem), resulting in additional mental changes."

According to Canamero [Cañamero 97], the principal advantages of emotions are:

- **Urgent take control.** In urgent situations such as danger because of fire for example, emotions take the control of the decision making in order to protect the virtual agents. "While motivations are in charge of driving behaviors under normal circumstances, emotions take over behavior control and changes goal priority in situations requiring an urgent response".

- **Re-equilibration of internal milieu** Emotions play a role in contributing to maintain the internal stability of the organism fundamental for adaptation. Although a homeostasic model is not enough to explain emotional states, emotions seem to play some role in homeostatis. Pribram [Pribram 84] attributes to motivational and emotional mechanism complementary roles. While motivation is concerned with the operations of appetitive processes that try to activate action as a response to deprivation, emotion is derived from processes that try to stop ongoing behavior, i.e., it is concerned with satiety processes of re-equilibration.

- **Management of social interactions.** Emotions and their expression are crucial in communication for autonomous agents situated in complex social environments. They give a lot of information about the current state of the virtual agents and increase the believability.

- **Emotions in mood and general state influences** Emotions are also related to moods and temperaments. A mood can be explained as a (low) tonic level of arousal of an emotional system, whereas emotions imply sudden activation and short duration. Temperaments can be explained as "predetermined" threshold levels that make the activation of a particular emotion more likely.

- **Make behaviors more realistic, autonomous, coherent and adaptive.** Emotions seem to have a central position in autonomy and adaptation in biological systems if we want to build better adapted and more "life-like" virtual characters with better communication capabilities, with more flexible behavior, showing human/animal-like types of "errors", etc.

But even thought motivations and emotions may play complementary roles with the respect to action selection, they cannot be placed at the same level [Frijda 95, Ortony 88, Rolls 99, Cañamero 98]. On the one hand, since an emotional system is a complex system connected to many other behavioral and cognitive subsystems, it can act on these other systems at different levels at the same time. On the other hand, since emotions are related with goals, they contribute to the generation of richer, more varied, and flexible behaviors in addition to motivations. Emotional processes are also relevant in 'higher' and 'lower levels' of an architecture and especially to link them thanks to hormonal system [Cañamero 01]. The separation of higher and lower levels is a helpful simplification. Operations at a lower or reactive level involve more direct coupling to an agent's environment, the time-scale of operations corresponds directly to the temporal resolution of the agent's interface to its world, and no symbolic representations need to be involved. Processes on higher levels use a representation of possible actions, or counterfactual reasoning that can detach representations from the current state of the world although the notion of representation has stirred many controversies in behavior-based AI [Brooks 91] and cognitive science [Clark 97].

Finally, emotions provide the primary blueprints for "cognition". Emotional states greatly influence perception and attention and also promote selective memory and learning [Bower 82]. [Tomkins 84] points out the generality of time, object, intensity, and density characteristic of emotions.

2.5.3 Mind

Minds can be thought as the control structures of autonomous agents. The function of a mind is at each instance to decide what to do next. Autonomous agents, and only autonomous agents, need minds. A mechanism of mind is some piece of the architecture of such a control structure that enables it to so decide [Franklin 95]. By building "entire" creatures we gain insight into the interrelationship between perception, behavior and action in a way that is difficult to achieve otherwise. Too often these issues are studied in isolation, despite the fact that they are highly interdependent in nature.

A fruitful approach would be to use the architecture to "build a mind" which was embodied in a creature and through that process begin to understand the strengths and weaknesses of the approach. Through this process we can gain understanding into the fundamental issues of action selection in animals, as well as a better understanding of our own minds [Blumberg 96]. Moreover motivations and emotions are a prerequisite for any cognitive systems [Sloman 87] and every design decision taken during an implementation constitutes a hypothesis about how human minds work [Franklin 97]. The concepts and methodologies of cognitive science and of computer science will work synergistically to enhance our understanding of mechanisms of mind.

2.6 Summary

Several domains have been a source of inspiration for modeling the action selection architecture for autonomous humans in persistent worlds. We conclude that they should have these requirements:

- **Situatedness**

  To respond quickly to the environmental changes such as opportunist behaviors

- **Pro-activeness**

  To manage autonomously the fulfillment of goals

- **Motivations**

  To give a "strong" autonomy to virtual humans by self-generating goals

- **Emotions**

  To modulate and evaluate the choice and to enhance social interactions

With these requirements fulfilled, virtual humans are highly autonomous and distinct. They can react differently to the same situations, because the decision-making process is individual. We claim that the individuality has to be modeled before considering social interactions in a "bottom-up" approach (see Section 2.1.1). The main goal of thesis is to understand individual action selection to obtain more complex and realistic autonomous virtual humans. This can be very useful for non player characters in video games so that they "live" their own life continuously without external interventions. In this case, the game will become more interesting.

Chapter 3

Related work on decision architectures

Postulated already by the first ethologists and observed in natural intelligence researches, hierarchical and fixed sequential orderings of actions into plans are necessary to reach specific goals and to obtain proactive and intelligent behaviors for complex autonomous agents [Bryson 00]. Hierarchies reduce the combinatorial complexity of action selection, i.e. the number of options that need to be evaluated when selecting the next action. Hierarchical systems are often criticized because of their rigid, predefined and unreactive behaviors [Maes 91, Brooks 91]. To obtain more reactive systems, constant parallel processing has been added [Tyrrell 93a, Blumberg 96].

However fully reactive architectures increase the complexity of action selection despite the use of hierarchies (see Section 2.4.3). Agent architectures should be both reactive and capable of complex tasks. If sequential and hierarchical controls are avoided, then chaining behavioral modules becomes difficult. Modularity in agent architectures simplifies their design process because it allows decomposing the agent's intelligence, or some parts of its intelligence, into a number of smaller, relatively autonomous units. These behavioral units can communicate with each other to decompose the whole problem in sub-problems. However agent architectures should monitor the environment so that the agent will choose appropriate responses according to unexpected and opportunist situations. To take advantage of both reactive and hierarchical systems, some authors have implemented an attention mechanism. It reduces the information to process and thus simplifies the task of action selection [Bryson 00, Thorisson 04]. These architectures with selective attention can surpass the performance of fully reactive systems, despite the loss of information and compromise behaviors [Bryson 00].

In our approach [Sevin 01], we choose to implement reactive and goal-oriented hierarchical classifier systems [Donnart 94] associated with the functionalities of a free flow hierarchy [Tyrrell 93a] for propagating the activity in order to respect the requirements for designing autonomous virtual humans in persistent worlds (see Section 2.6).

3.1 Hierarchical classifier systems

3.1.1 Traditional classifier system

Classifier systems [Holland 75, Wilson 87] are good tools to classify binary or continuous input data into common categories. They also can be used as decision-making mechanisms for designing action selection mechanism of virtual agents to choose what actions come next.

Classifier systems have generally four modules:

1. **Input interface**

   This module transforms input data coming from the sensors into sense messages sent to the messages list

2. **Output interface**

   This module transforms action messages coming from the messages list into actions sent to the effectors.

3. **Message list**

   This module manages three types of messages:

   - sense messages coming from the input interface
   - action messages sending to the output interface
   - internal messages staying in the message list during one or more iterations in order to activate other classifiers.

4. **Rule base**

   This module contains all the production rules or classifiers written according to this template:

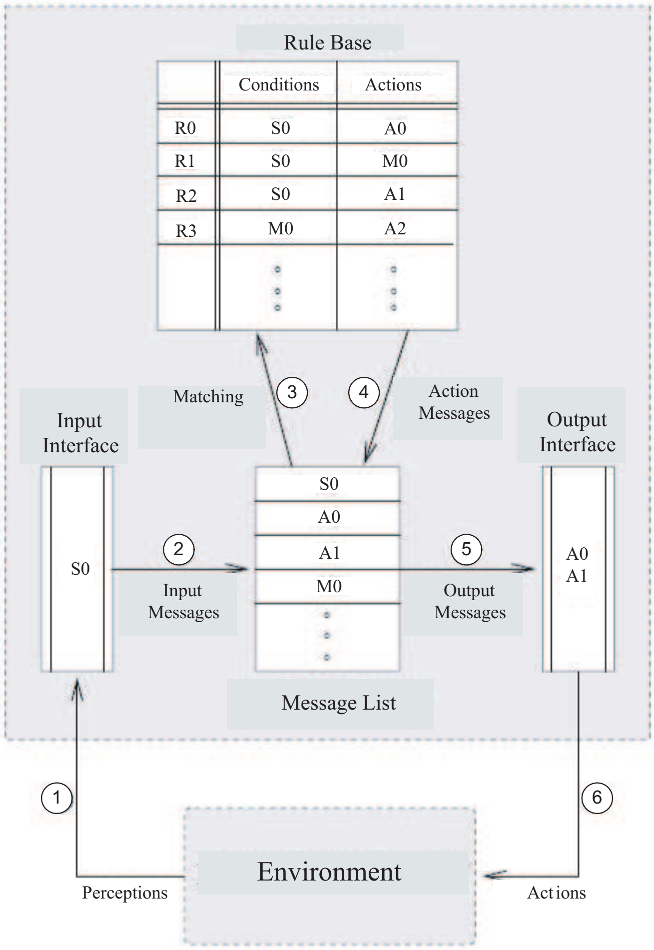

Figure 3.1: General description of a classifier system.

- *internal rules*

  If sense message = condition rule **then** put the internal message on the message list

  If internal message = condition rule **then** put the internal message on the message list

- *external rules*

  If sense message = condition rule **then** put the action message on the message list

  If internal message = condition rule **then** put the action message on the message list

Classifiers have weights generated during the transformation of the action messages into actions and these weights are useful to choose the best action when several ones are possible for effectors. Learning algorithms such as Profit sharing plan [Grefenstette 88] or bucket brigade [Holland 86] can be used to update the classifier weights according to the consequences at short or long term and new rules can be create with genetic algorithms. Classifier systems manage mostly reactive behaviors such as a quick reaction to a change in the environment. For managing more cognitive behaviors with a classifier system, one needs to add internal messages. A sequence of actions will be generated even if nothing changes in the environment. This process can be used to implement internal reasoning and to choose the best action according to the long term consequences. However this functionality is not often used because of the difficulty to design it in particular with genetic algorithms [Donnart 94].

3.1.2 Definition of hierarchical classifier systems

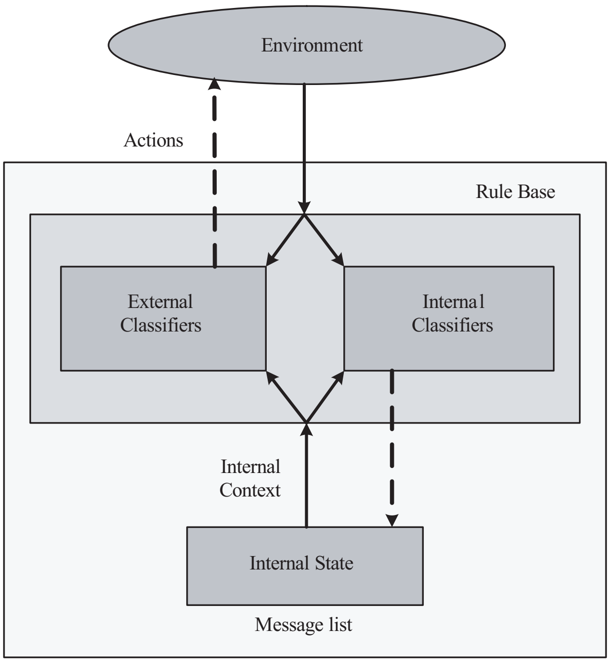

Figure 3.2: General description of a hierarchical classifier system.

Hierarchical classifier systems [Donnart 94] provide a good solution for modeling complex systems by reducing the search domain of the problem. A classical classifier system has been modified in order to obtain a hierarchical organization of the rules, instead of a sequential one (see Figure 3.2). A hierarchical classifier system can generate reactive as well as goal-oriented behaviors because two types of rules exist in the rule base: the external ones, to send actions directly to the motors and the internal ones, to modify the internal state of the classifier system. The message list contains only internal messages, creating the internal state of the system, which provides an internal context for the activation of the rules. The number of matching rules is therefore reduced, as only two conditions need to be fulfilled to activate rules in the base: the environmental information and the internal context of the system. Internal messages can be stacked in the message list until they have been carried out by specific actions. Then behavioral sequences of actions can easily be performed. For example, the virtual human can move to a specific location from anywhere with the aim of performing actions that will satisfy his motivations. Figure 3.4 and Figure 3.5 show how a hierarchical classifier system can generate such a sequence of actions to satisfy hunger.

3.1.3 Message list modification - internal state

In hierarchical classifier systems, the message list contains only internal messages (see Figure 3.2). Action messages are directly sent to the effectors after arbitration if some actions are conflicting. The internal state of a classifier system corresponds to the internal messages in the message list. It gives a specific internal context to activate the rules in the message list. The internal context changes when an internal message is added or deleted. The internal messages can stay long in the message list but they should have a limited life span thanks to canceling conditions.

3.1.4 Rule base modification - external and internal rules

The structure of the classifiers in the message list is modified to take into account the internal context in the activation of the rules. Each classifier is composed of two conditions. These two conditions must be satisfied to activate the classifier:

- *First condition* should match with messages coming from the environment information
- *Second condition* should match with the context generated by the internal state

With this structure modification, hierarchical classifier systems have two types of rules (see Figure 3.3):

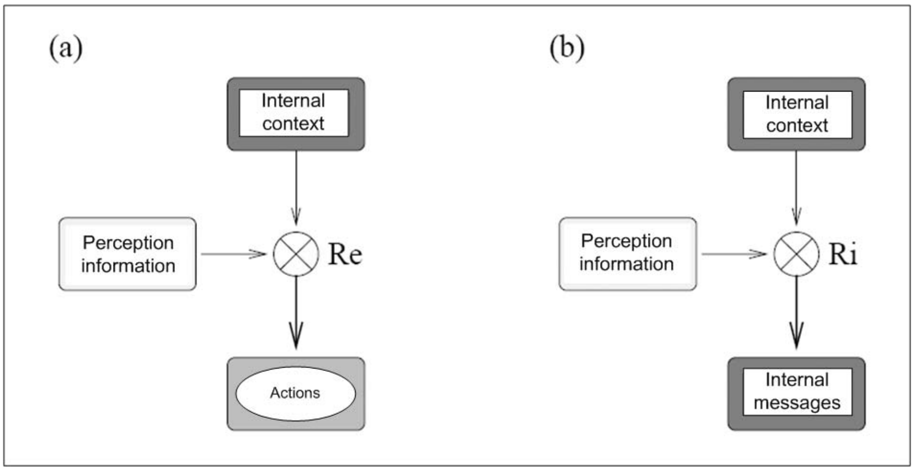

Figure 3.3: Schematic view of the activation of rules: (a) external rules and (b) internal rules

- *External rule or classifiers* suggest actions to send to effectors. They can be expressed like this:

  If "perceptual information = first condition of the rule" **and** "Internal context = second condition of the rule" **then** "send action to effectors"

- *Internal rules or classifiers* modify the internal state of the classifier system by putting internal messages in the message list. They should match with the context generated by the internal state and can be expressed like this:

  If "perceptual information = first condition of the rule" **and** "Internal context = second condition of the rule" **then** "put the message in the message list".

3.1.5 Hierarchical organization

Sequences of behaviors can be described by hierarchical classifier systems as shown this following example: the behavioral sequence of actions for "feeding" needs one internal classifiers (R_1) and three external classifiers (R_2, R_3 and R_4):

R_1**:** if "visible food" and "feeding", then "take food".

R_2**:** if "distant food" and "take food", then "reach food".

R_3**:** if "near food" and "take food", then "hold food near mouth".

R_4**:** if "food near mouth" and "feeding", then "eat".

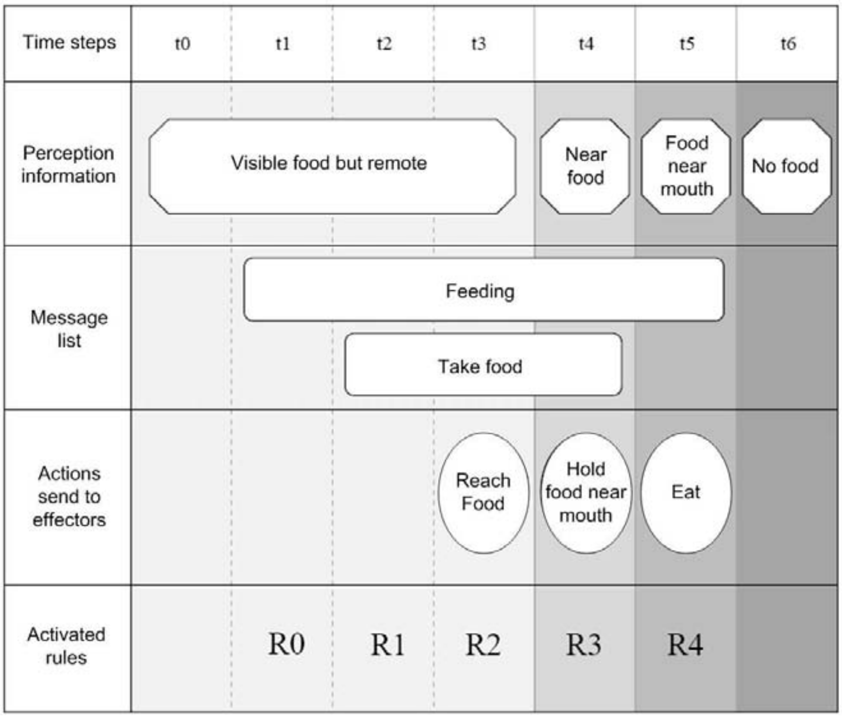

Figure 3.4: Example for generating a sequence of actions using a hierarchical classifier system (timeline view)

The canceling conditions of the internal messages "feeding" and "take food" from the message list can be associated respectively with "no visible food" and "food near mouth". Figure 3.4 and Figure 3.5 shows how the rules are activated according to the perception information and the current state of the hierarchical classifier system (internal context) for the same example.

The virtual character sees some food and wants to eat it. Then, at stage t_0 of Figure 3.4 the internal message "feeding" is put in the message list. As the food is visible but remote, the second internal message "take food" is put in the message list at stage t_1 thanks to the internal rules R_0. This internal message "take food", is used as the internal context to activate first the external rules R_1 at stage t_2 according to the perception information. The respective action "reach food" is proposed as a possible action for the effectors and is executed as there are no other conflicting actions modifying its environment. When the virtual character can take the food, at stage t_3, the external rule R_2 is activated according to the internal context of the hierarchical classifier system: "take food" and the perception of the virtual character "near food". As the virtual character is situated, it perceives the environmental changes corresponding to the canceling conditions of the internal message: when the virtual character senses that it holds the food near the mouth, the internal message "take food" is cancelled and the last external rule R_3 is activated at stage t_4 because both conditions are satisfied. In this case, "feeding" is used as the internal context and the virtual character can eat the food. When it perceives there is no food, the internal message "feeding" is cancelled from the message list at stage t_6.

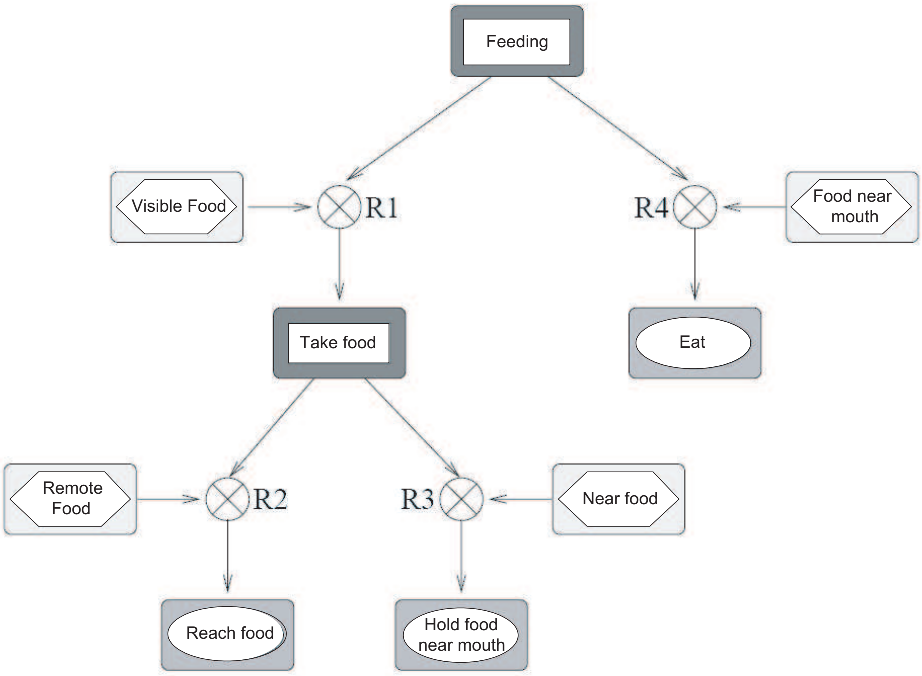

Figure 3.5: Generating a sequence of actions using a hierarchical classifier system (hierarchical view).

Figure 3.5 shows the same example but with a hierarchical view instead of a timeline view. The two conditions for activating rules and the hierarchical organization of the behavior are better represented. Thanks to the hierarchical classifier systems, a sequence of actions can be generated to do a specific task "feeding".

3.1.6 Hierarchical classifier system advantages

Simplification of the rule encoding

Hierarchical classifier systems encode more easily the rule-based systems in which many combinations of conditions must be satisfied to activate an action. For instance, the user

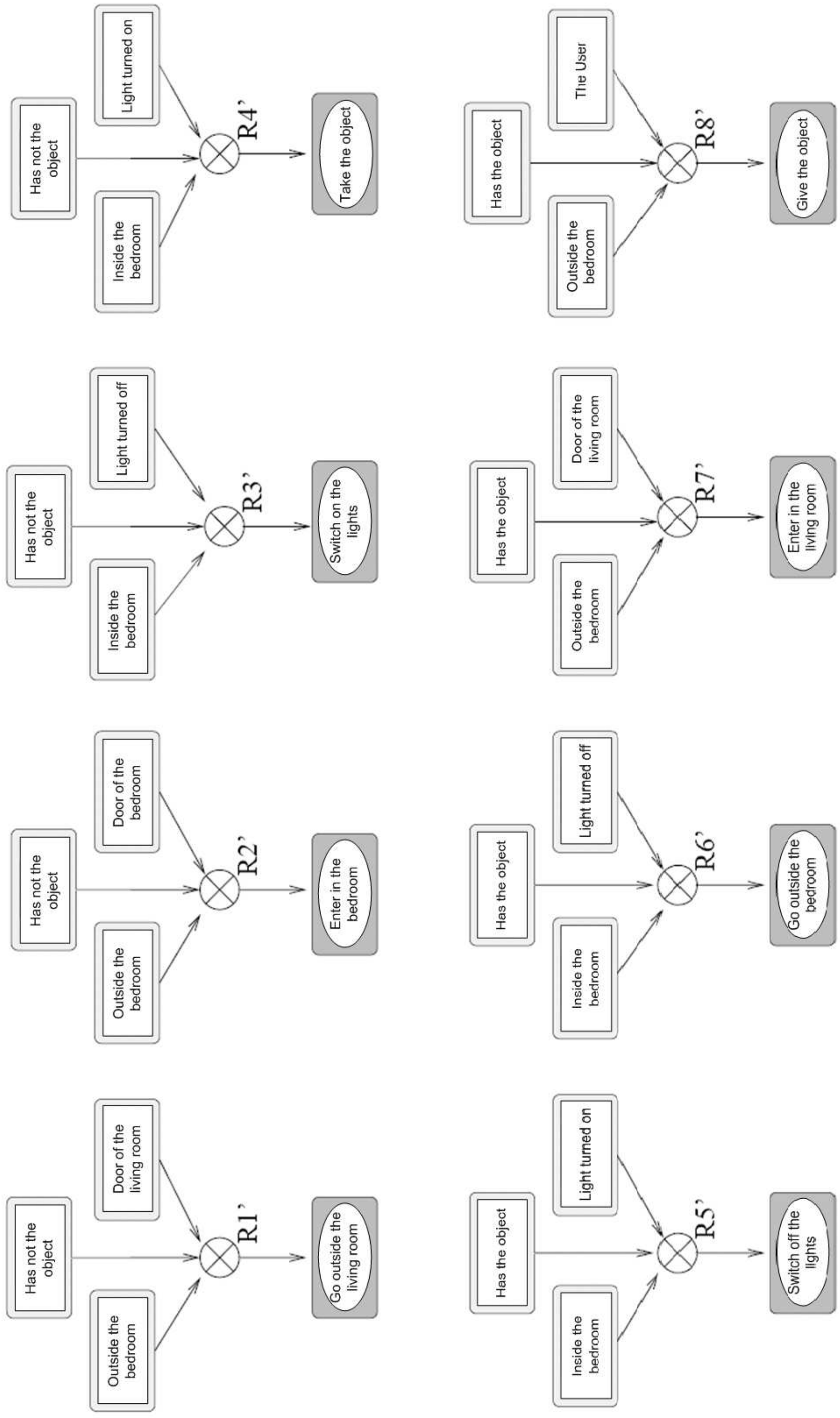

Figure 3.6: Description of the sequence of elementary actions with a classical classifier system. The virtual character should bring back an object to the user. The R_i' represent the rules used by the system for generating the sequence of actions. The ellipses are actions and the hexagon perception information

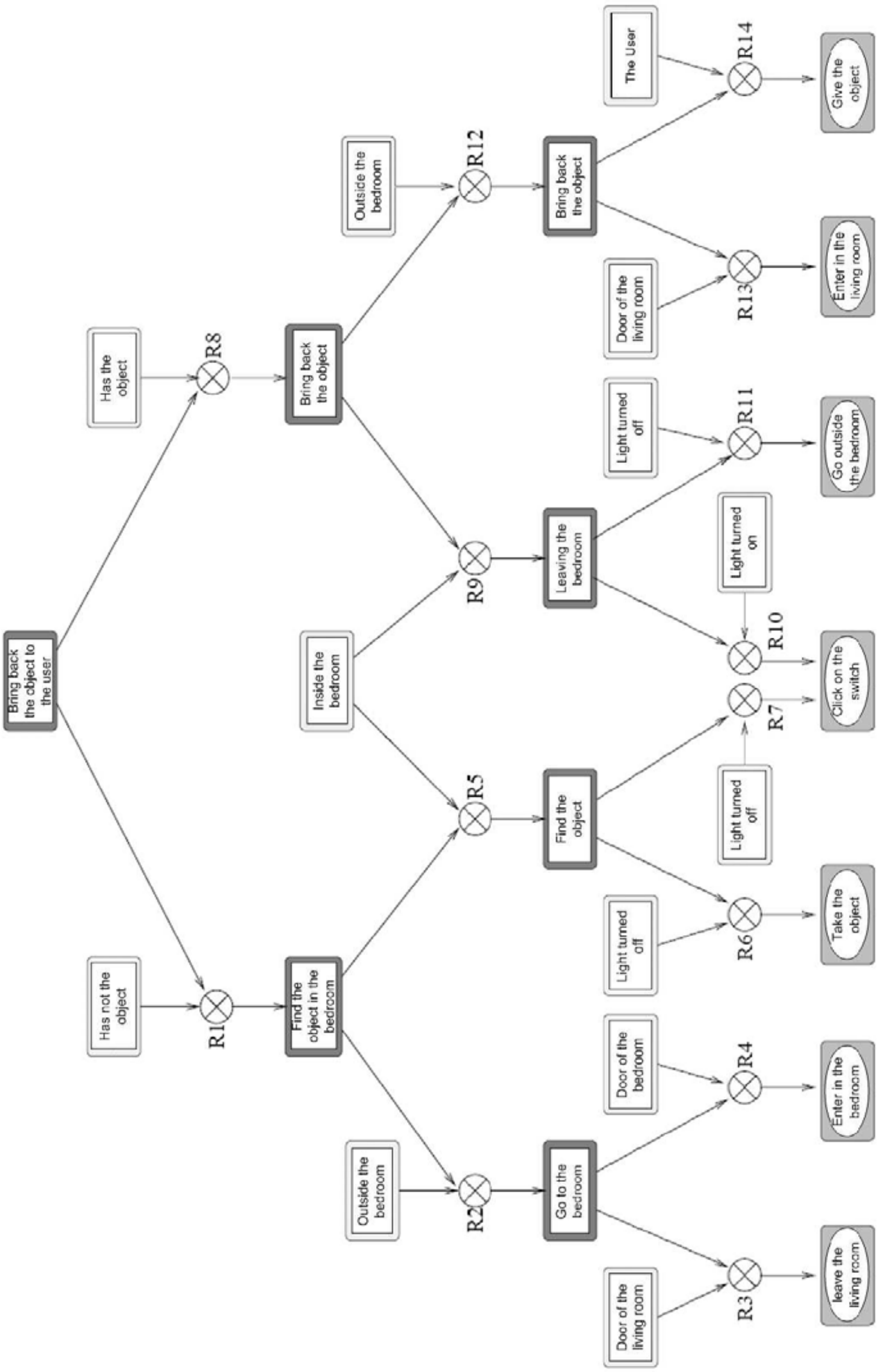

Figure 3.7: Description of the sequence of elementary actions with a hierarchical classifier system. The virtual character should bring back an object to the user. The R_i' represent the rules used by the system for generating the sequence of actions. They can be internal or external. The black squares are internal messages, the white squares are perception information and the ellipses are actions.

interacts with the virtual character in its virtual environment and asks it to bring back an object in the bed-room. As it is situated in the living-room, the virtual character has to "get out of the living-room", "enter in the bed-room", "switch on the light", "take the object", "switch off the light", "get out of the bed-room", "enter the living-room", and at last "give the object to the user". Traditional classifier system can be used for designing this sequence of elementary actions. As it is depicted in Figure 3.6, eight rules are necessary and three perception conditions should be satisfied for each activating rule.

However, the behavior "bring back an object" can be decomposed hierarchically into six internal rules and eight external rules (see Figure 3.7). The disadvantage of coding six extra rules is balanced with two important advantages:

- *one perception condition:* each rule should match with only one perception condition and one internal context condition whereas in a classical classifier system three perception conditions are necessary. For instance, the action "take the object" is activated only if the light is switched on according to the internal context "find the object". In the traditional classifier system, for activating the same action, three perception conditions should be matched: "inside the bedroom", "has not the object" and "light turned on".

- *reduction of the rule matching number:* with hierarchical classifier systems, the number of matches for generating a sequence of atomic actions is reduced. The hierarchy can be organized to test only once the perception conditions. For example, the perception "has not the object" is tested only with the R1 rule in the hierarchical classifier system instead of 4 times (R_1, R_2, R_3 and R_4) in a traditional classifier system. For this sequence and with the hierarchical classifier system, only 14 matches are necessary instead of 24 in the traditional classifier system.

Learning

Donnart in his Monalysa (Motivationally autonomous animat) architecture [Donnart 96c] has used efficient learning using two types of reinforcement on two different weights for each rule (for more details see [Donnart 96b]):

- *Internal reinforcement applied on local weights:* estimate the interest of the rules in their internal context linked to the internal messages thanks to a traditional Profit sharing plan. This is added to take into account the sub-behaviors decompositions

- *External reinforcement applied on global weights:* estimate the interest of the rules in action sequences. Donnart used a hierarchical Profit sharing plan for taking into the participation of the rules the decomposition of the behaviors in sub-behaviors.

Hierarchical classifier systems are more reliable and efficient than classical classifier systems [Donnart 94]. Moreover they allow complex task learning in a reasonable time. However, the designer has to pre-implement behavior hierarchies and to define the internal reinforcement for each behavior unit of the hierarchy. It implies that the designer should know the problem, the solution, and decompose it recursively into sub-problems. In this case, the system has to learn the rule weights to find the most adequate decomposition but cannot create new rules due to the hierarchy organization. Donnart [Donnart 96a] has implemented a mechanism for creating new internal rules that activate new internal messages, changing the hierarchy structure. Thus, the architecture is more adaptive, because it is not fixed. This mechanism is only used when is estimated more efficient, thanks to a heuristic cost function defined for each behavior unit and in parallel with existent possible behavior decompositions. The definition of the heuristic cost function is strongly dependent on the application and should be pre-implemented by the designer. Donnart [Donnart 96a] has used hierarchical classifier systems for constructing automatically a hierarchy of action plans that can be activated directly by the animat in simulation and in robotics. He uses also hierarchical classifier systems for discovering hierarchical representations of the environment where the animat is situated.

Reduction of the problem search space

Whereas the number of rules is more important in hierarchical classifier systems, the hierarchical organization of the problem reduces the size of the space for representing the whole problem. By decomposing the whole problem in sub-problems, the size of search for each sub-problem is smaller. With a recombination of the sub-problems, the solution of the whole problem is obtained. For example, the problem "feeding" involves solving the sub-problem "take food". This decreases the complexity of the problem. Learning is then more efficient and fast, but it requires an intermediate reinforcement mechanism for each sub-problem in addition to the main problem.

Creating new rules

Genetic algorithms can be used to create new internal and external rules from the existent ones as in classical classifier systems. As the size of the problem search space is reduced with the decomposition of the problem into sub-problems, genetic algorithms can find rules to solve new problems inside the sub-problems space. Then, the process of creating new rules is less random than in a classical classifier system. Moreover, the created rules are similar to those in the same sub-problem because the creation is limited to this sub-problem. In the end, it is quicker and more efficient to find general rules for solving each sub-problem instead of the whole problem.

Cognitive mechanisms

Classifier systems can have cognitive mechanisms which need reasoning on internal knowledge only if they are able to manage internal messages in the message list. As the principal goal of hierarchical classifier systems is to optimize the use of internal messages, there are very appropriate to manage cognitive mechanisms. They have two advantages compared to the classical classifier systems:

- *short term memory:* corresponds to the message list where the internal state of the hierarchical classifier systems is memorized for generating the internal context in order to activate the rules. The internal state is used to store where the virtual character is in its environment, its goals, its motivations and the current decomposition of action sequences.

- *long term memory:* corresponds to the rule base where external and internal rules are memorized. External rules activate actions according to perception information and the internal context of the hierarchical classifier systems. These rules can be inside sequences of actions or reflex, e.g., if the virtual character detects something in its environment, external rules activate the appropriate reflex actions. Internal rules are used to represent more general strategies of behaviors such as the action plans.

In artificial intelligence, the generation of plans is a slow, not adaptive mechanism. With a hierarchical classifier system and external and internal reinforcement learning, the process is improved.

Situated planning

In artificial intelligence, plan generators construct action plans. These plans are a sequence of operators used for reaching a specific goal. The plan generator simulates the execution of the operators using a precise representation of the current environment until it finds the solution. The efficiency of this method is in the quality of the world model. It should be pre-implemented by the designer [Agre 90] and as perfect as possible. In this case, the simulation will be identical to the results of the execution. No interaction with the environment is necessary. On the contrary, hierarchical classifier systems are situated systems. It has several advantages compared to the internal reasoning mechanisms in artificial intelligence:

- quicker
- more adaptive
- more opportunistic

The situated systems have several decision modules using a limited knowledge of the environment and each carries out rather simple behaviors [Maes 92]. However, the combination of the modules can achieve complex behaviors emerging from the module interactions. Situated systems are simpler to implement and more adaptive to the environment because they don't use global world model but rather the current environment the world model has. Moreover hierarchical classifier systems can also decide in the long term, not only in the short term, thanks to the internal messages in the message list. The rules are pre-implemented by the designers but the activation of the rules and the decomposition in sub-problems is due to the interaction between the virtual character and the environment. The activation of plans and their decomposition depend on the perceived situation. Hierarchical classifier systems are able to do a situated planning that can be optimized by learning.

Motivated and motivationally autonomous agents

Action selection depends on motivational systems [Meyer 97]. Based on ethological studies, McFarland [McFarland 91, McFarland 93] has defined a typology that ranges from simple motivated automata to motivationally autonomous agents (see Figure 3.8). Motivated automata allow the agent choosing the most appropriate action according to its perception of the environment, its physiological states, and the short term consequences of the chosen action. Motivationally autonomous agents are the same but they also take into account the short and long term consequences of all the possible actions, the motivations and the evaluations of the decision. An animal can act sometimes as a motivated automaton and sometimes as a motivationally autonomous agent.

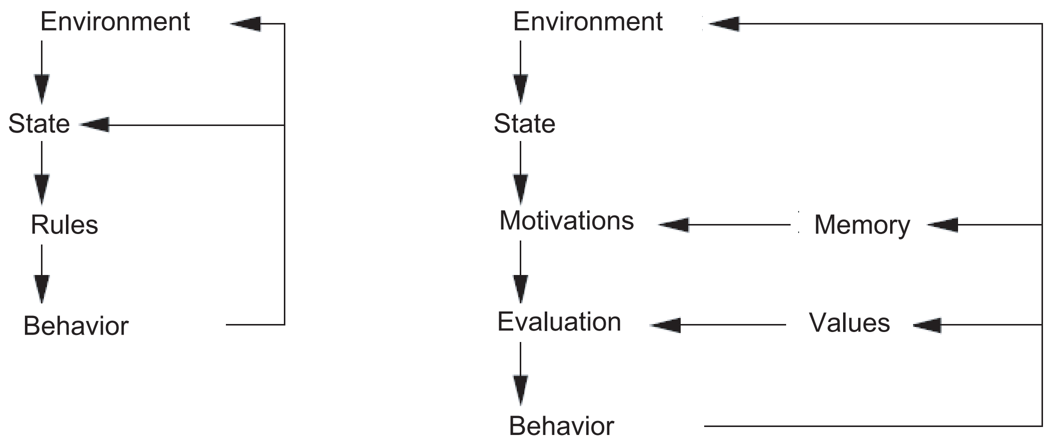

Figure 3.8: Differences between a motivated automaton and a motivationally autonomous agent. [Donnart 98]

Hierarchical classifier systems can span between these two types of agents. External rules can activate actions according to their perception and internal context. For instance, if there is some danger, the agents react accordingly because of their fear and the perceived danger without reasoning. In this case, the hierarchical classifier systems are comparable to motivated automata. However, when internal rules are activated, they put internal messages that will not have immediate consequences. These behaviors need several elementary actions to be realized. For example, "bring back an object" implies many elementary actions such as "find the object", "take the object", etc. In this case, hierarchical classifier systems are comparable to motivationally autonomous agents.

3.2 Free flow hierarchy

3.2.1 Definition

An attribute common to many action selection mechanisms is the use of a winner-takes-all selection/arbitration process, where the final decision is made exclusively by the winning action or behavior [Tu 99]. While this offers highly focused attention and hence efficiency, it ignores the importance of generating compromise actions, i.e., satisfying many motivations at the same time. The ability to compromise between different, even conflicting desires is evident for natural animals and very important in ethology. Tyrrell [Tyrrell 92] emphasize this particular aspect of animal behavior in the implementation of what is known as free-flow hierarchies. Free-flow architectures are first proposed by Rosenblatt and Payton [Rosenblatt 88]. A free-flow hierarchy implements compromised actions within a hierarchical action selection architecture similar to those proposed by early ethologists such as Tinbergen [Tinbergen 51].

3.2.2 Activity propagation

The hierarchy is composed of nodes which receive information from internal and external stimuli in the form of activity. The nodes feed their activity down through the hierarchy until those arrive at the action nodes, i.e., the leaves of the tree, where a winner-takes-it-all process decides which action is selected (see Figure 3.9). It violates the subsumption philosophy [Brooks 86] and allows sums of activities. In classical hierarchies, the choice of the most activated node is made at each level.

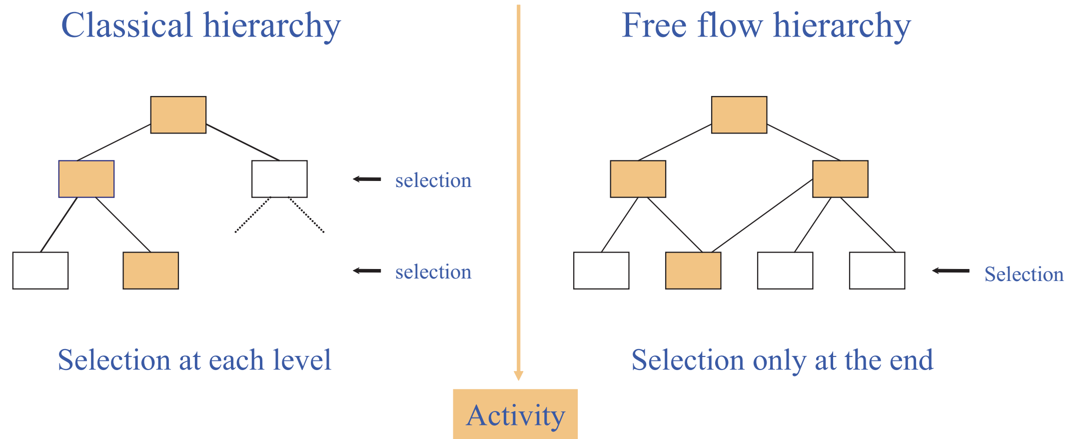

Figure 3.9: Comparison of the activity propagation between classical hierarchy and free flow hierarchy

Free flow hierarchies increase the reactivity and the flexibility of the hierarchical systems because of this unrestricted flow of information, the combination of preferences and the possibility of compromise and opportunist candidates. All these functionalities are necessary to select the most appropriate action at each moment.

3.2.3 Compromise behaviors

"Compromise Candidates can be defined as the need to be able to choose actions that, while not the best choice for any one sub-problem alone, are best when all sub-problems are considered simultaneously." [Tyrrell 93a] (see Section 2.1.3). Biologists have noted apparent compromise behavior among animals selecting between multiple targets [Morris 78, Latimer 87, Bailey 90]. According to Lorentz [Lorenz 81], in real life, humans or animals have a lot of motivations to satisfy at the same time and in the end it is often compromise behaviors that are chosen because they better maintain the homeostasis of the need, decrease the risk of oscillations and increase the flexibility of the action selection mechanism. The ability to consider compromise actions in an uncertain world makes great intuitive sense. When multiple goals interact, solving each optimally is not always optimal for the overall system. From the perspective of an virtual character in an environment where desired resources might go away at a moment's notice, a compromise movement in a direction in-between the resources might increase the virtual character's likelihood of obtaining one resource in the event that the other becomes unavailable [Crabbe 02].

3.2.4 Free-flow hierarchy advantages

According to Steegmans [Steegmans 04], the main advantages of free-flow architectures are the following:

- Stimuli can be added to the relevant nodes avoiding the 'sensory bottleneck' problem: in a hierarchical decision structure, to make correct initial decisions, the top level has to process most of the sensory information relevant to the lower layers.
- Decisions are made only at the level of the action nodes. Thus all available information is taken into account to select actions.
- Since all the information is processed in parallel the agent can take different preferences simultaneously in consideration. E.g. consider an agent that moves to a spotted object but is faced with a neighbouring threat. If the agent is only able to take into account one preference at a time it will move straight to the spotted object or move away from the threat. With a free-flow decision tree the agent avoids the threat while it keeps moving towards the desired object, i.e. the agent likely moves around the threat towards a spotted object.

3.2.5 Tyrrell's test

In his PhD thesis, T. Tyrrell [Tyrrell 93a] demonstrated that hierarchical free-flow architectures are superior to flat decision structures, especially in complex and dynamic environments. He tests four well-known action-selection mechanisms coming from ethology

[Lorenz73, Baerends76], Maes's behavior-based network [Maes 90] and two types of free flow hierarchies (the original one [Rosenblatt 88] and Tyrrell's modified one [Tyrrell 93b]) in an artificial life environment with many conflicting motivations.

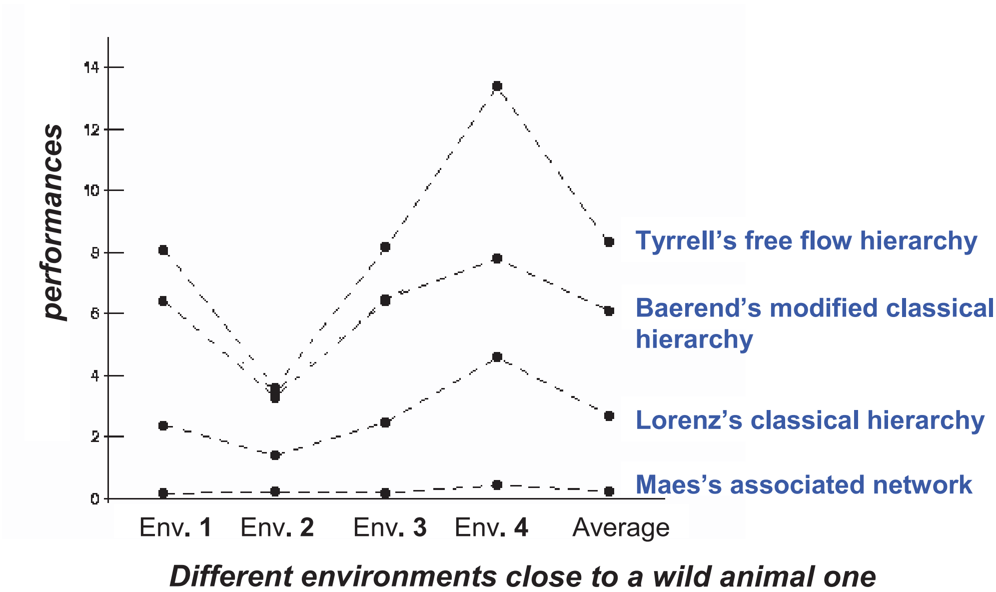

Figure 3.10: Tyrrell's test results [Tyrrell 93a]

In this environment, a small animal needs to balance a large number of often conflicting goals of very different types and to deal with predators and limited quantity of resources. This involves six types of subproblems [Bryson 04]:

- Finding sustenance. This includes water and three forms of nutrition, which are satisfied in varying degrees by three different types of food.
- Escaping predators. There are feline and avian predators, which have different perceptual capabilities and hunting strategies.
- Avoiding hazards. Latent dangers in the environment include wandering herds of ungulates, cliffs, poisonous food and water, temperature extremes and periodic (nightly) darkness. The environment also provides various forms of shelter including trees, grass, and a den.
- Grooming. Grooming is necessary for homeostatic temperature control and maintaining general health.

- Sleeping. The animal is blind at night and needs to sleep to maintain its health.
- Reproduction. The animal is male, thus its reproductive task is reduced to finding, courting and inseminating mates. Attempting to inseminate unreceptive mates is hazardous.

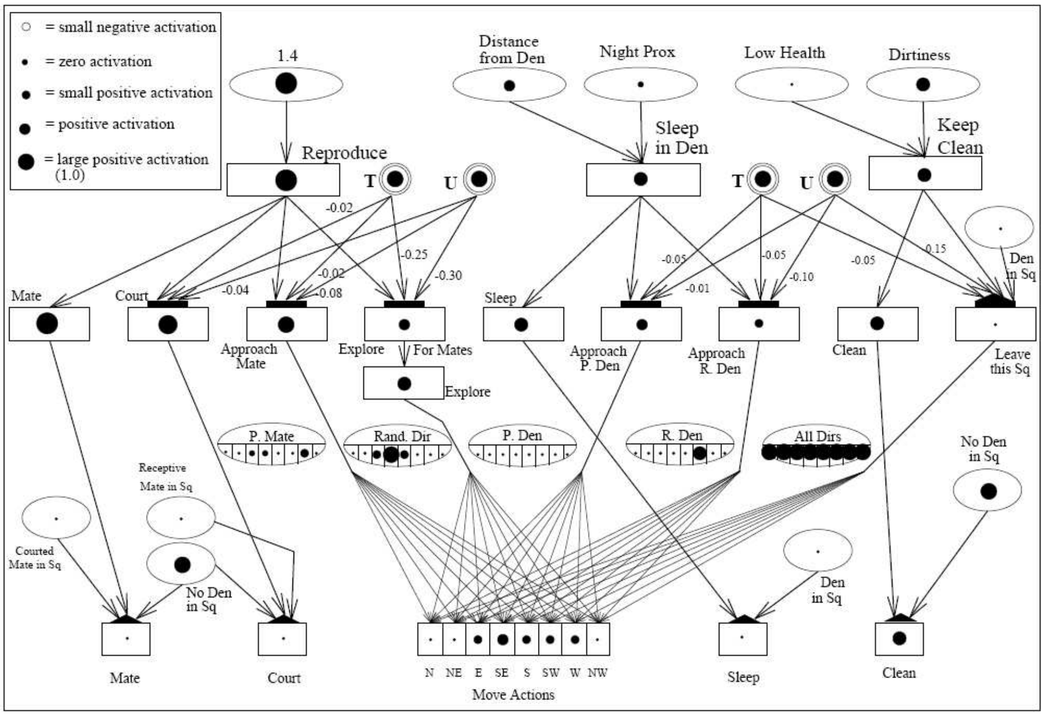

Figure 3.11: A fraction of the Extended Rosenblatt & Payton action selection network for controlling an animal in Tyrrell's Simulated Environment. T and U are temporal and uncertainty penalties for actions that take time or pursue a goal uncertain to succeed. P stands for perceived, R for remembered [Tyrrell 93a].

Using extensive experimentation (simulations cover up to 10 days of life and involve thousands of decision cycles per day), Tyrrell demonstrates a substantial advantage for all of the hierarchical architectures he modeled over Maes's approach and its nearest strictly-hierarchical competitors because it is able to take into account the needs of all behaviors [Bryson 00]. He claims that free flow hierarchy is the best solution for decision architecture when managing many conflicting motivations and goals. Tyrrell's work has had significant impact on the Action Selection field [Humphrys 96, Decugis 98, Bryson 00, Girard 02], and a number of researchers have developed systems to meet the criteria he set out [Werner 94, Blumberg 94, Crabbe 02] (see Section 3.2.6).

3.2.6 Requirements for action selection mechanisms

Finally the requirements of Tyrrell [Tyrrell 93a] for designing a mechanism of action selection can be resumed to six essential criteria:

- Taking motivations into account.
- Taking environment information into account.
- Preferring to perform motivated actions (eat, drink...) over locomotion actions (go East, South...).
- Carrying out the current sequence of actions to its end in order to satisfy the current motivation (generation of behavior sequences of locomotion actions and persistence of motivated actions).
- Interrupting the current sequence of actions if another motivation becomes higher or if opportunist behaviors occur, and switching to a new behavior to satisfy the new motivation.
- Preferring compromise behaviors, i.e. where the chosen action satisfies the greatest number of motivations.

3.3 Summary

The efficient action selection architecture will associate hierarchical and reactive systems with no loss of information and compromise behaviors. Our action selection model for autonomous virtual humans is based on hierarchical classifier systems which can respond to environmental changes rapidly with its external rules and generate situated and coherent behavior plans with internal rules. The problem complexity is also reduced because only two conditions need to be fulfilled: the internal context of the hierarchical classifier systems and the perceptions. Hierarchical classifier systems are adapted for designing autonomous virtual humans because they have cognitive capacities thanks to their short and long term memories. Our model is also based on free flow hierarchy to allow compromise and opportunist behaviors increasing the reactivity and the flexibility of hierarchical systems. It respects ethological criteria for designing efficient action selection mechanisms.

Chapter 4

The Model of Action Selection for Autonomous Virtual Humans

With the constraints of real-time simulations, the notion of representations and database of knowledge are avoided. They are used in cognitive agent architectures such as SOAR [Laird 87] or ACTR [Anderson 93] and have stirred many controversies in behavior-based AI [Brooks 91] and cognitive science [Clark 97]. The situated and bottom-up approach where the agent is in permanent interaction with the environment is necessary to have coherent decision-making in real-time. However hierarchical and fixed sequential orderings of actions into plans are necessary to reach specific goals and to obtain proactive and intelligent behaviors for complex autonomous agents. In our approach [Sevin 01], we choose to design the action selection model (see Section 2.1.2) to use reactive and goal-oriented hierarchical classifier systems [Donnart 94] associated with the functionalities of a free flow hierarchy [Tyrrell 93a] for the propagation of the activity. Hierarchical classifier systems (see Section 3.1) allow to have a coherent and robust behavior by finding sequence of actions in order to achieve goals. They can also react quickly to the environmental changes. The free flow hierarchy (see Section 3.2) brings reactivity and flexibility to the hierarchical system and allows to have no loss of information and to achieve compromise behaviors necessary for effective action selection mechanisms.

4.1 Architecture levels

In a "bottom-up" approach (see Section 2.1.1), we follow the point of view of many ethologists: try to understand simple animals functioning in order to understand more complex ones based on an evolutionary process. We follow this approach by first designing a fully reactive system and then by adding pro-active, motivational, and emotional levels (see Section 2.6) in order to have more interesting and believable autonomous virtual humans.

4.1.1 Reactive level

The first level of our decision-making architecture is reactive so that it reacts accordingly to the changes in the environment. The virtual human is situated (see Section 2.4.2) in his environment, i.e., he perceives it and acts on it in a perception-action loop (see Figure 4.1).

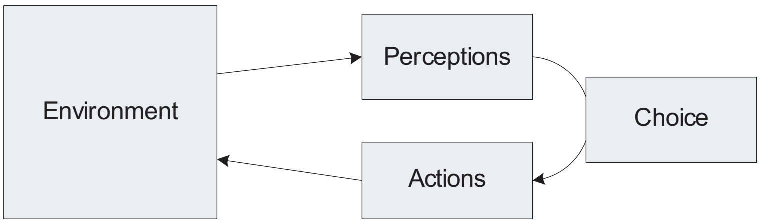

Figure 4.1: Schematic view of the reactive level

Environmental perceptions increase the notion of individuality, thus they are specific for each virtual human, who interprets them depending on the current situation and his own experiences. For instance, if the virtual human is hungry, he will search for food sources in the environment to satisfy his hunger. Brooks [Brooks 91] was the first in robotics to claim that intelligence should be designed depending on the environment and not on the predefined representations in a database.

Hierarchical classifier systems manage reactive response according to the perceptions thanks to their external rules (see Section 3.1.4). These rules match directly the perceptions and send actions to effectors. Then the action selection mechanism should choose the most adequate one. It allows opportunist behaviors (see Section 4.2.2) which are essential in action selection mechanisms. For instance, a virtual human is a little more thirsty than hungry and goes to the drink source. If on his way he finds a food source, he should stop to eat instead of continuing his way to the drink source and come back for food later. This is common sense and it is proven in ethology that animals act this way, because it is better for their survival. The behavior is adapted to the situation with a mechanism that increases the urgency to satisfy the motivations according to the level of the need.

The free flow hierarchy amplifies the importance of the environment in the decision-making system because no decision is made before the lowest level of the hierarchy (see Section 3.2.2). Then, all the possible actions are kept in the propagation of the activity in the system and it is only in the end that the system chooses one action. In this case, compromise behaviors that can satisfy several motivations at the same time (see Section 4.2.6) are possible. In real life, humans or animals have a lot of motivations to satisfy at the same time and finally it is often compromise behaviors that are chosen because they better maintain the homeostasis of the need, decrease the risk of oscillations, and increase the flexibility of the action selection mechanism.

A reactive level is a prerequisite for designing an effective action selection model in real-time for autonomous virtual humans (see Section 2.4.2). A fully reactive action selection model is situated and well adapted to the environmental changes. However, virtual humans do not simply act in response to their environment; they have to be able to exhibit goal-directed behaviors by taking the initiative, when appropriate.

4.1.2 Pro-active level

A pro-active level with a behavioral planner (see Figure 4.2) is necessary for obtaining ordered sequences of actions to execute goal-oriented behaviors such as "go to the kitchen and eat" (see Section 3.1.5). Indeed, the virtual human has to go to the kitchen from where he stands and once he is in the kitchen, he has to find the food and put it in his mouth. This sequence of actions is not easy to achieve, because it implies to solve many problems such as avoiding obstacles, manipulating and grasping objects, etc...

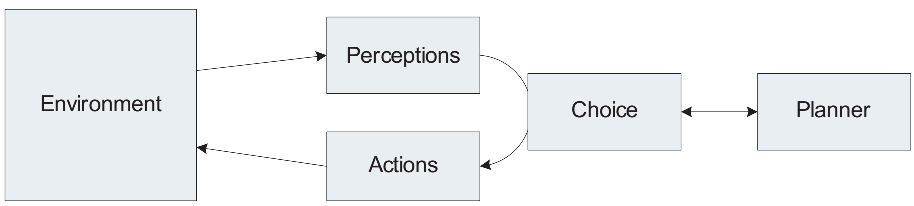

Figure 4.2: Schematic view of the pro-active level

Hierarchical classifier systems can manage easily the planning necessary for fulfilling specific goals, thanks to the internal rules (see Section 3.1.4). In his thesis, Donnart [Donnart 98] has shown the capabilities of hierarchical classifier systems in behavioral planning (see Section 3.1.6). Internal messages are sent to the message list and they create an internal context for activating other internal or external rules. With this technique, a coherent sequence of actions such as: "reach a place where the food is, take the food and put it in the mouth" can be generated according to the environment (see Section 4.2.5).

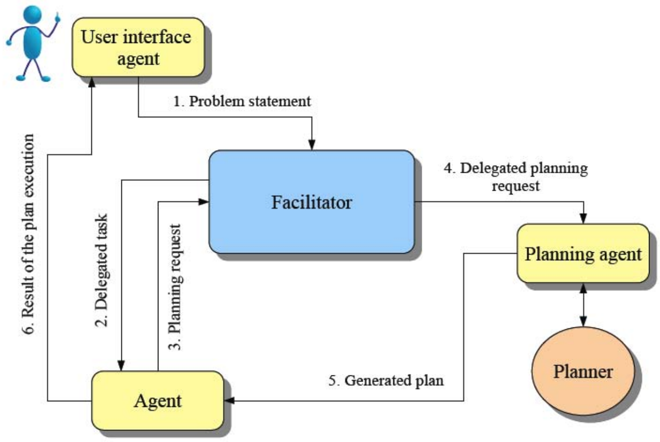

Figure 4.3: Diagram of the planning process

We choose to use only the capabilities of the hierarchical classifier systems for the action plans and keyframes for executing motivated actions in order to focus on the decision making. However in the VRLab, works on more complex planners [Ciger 05] and smart objects [Kallmann 01, Kallmann 03a] are done. More complex planners can solve general actions such as "buy 2 train tickets for London" and find a solution inside a rule base to accomplish them. With the smart objects, many possibilities of interactions between the virtual humans and the objects are possible because they contain all the necessary information for using them, such as the place where the hands should be placed to grasp the object, the possible movements that the virtual human can do with the object, etc, avoiding the use of representation databases. However, it requires time to tune these smart objects while keyframes are generated rapidly. Such a complex planner and smart objects could be integrated in our system in the thanks to its modularity.

These sequences of actions can be interrupted at any moment because one motivation becomes more urgent to satisfy or opportunist behaviors occur (see Section 4.2.7). In our model, the coherence is maintained because the highest motivation sends its activities to the system and in the end; the corresponding action has a great chance to be the most activated one. In this case, the action selection mechanism chooses it and the action sequence will be performed until t the associated internal variable is decreased. However, at each iteration, the choice is made at the lowest level of the hierarchy (action level)

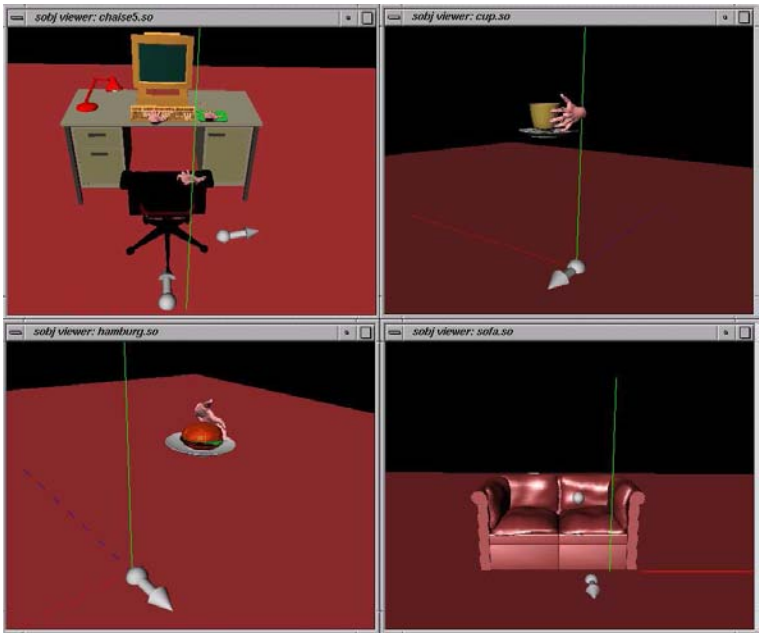

Figure 4.4: Examples of actions modeled with smart objects

according to the free flow hierarchy. In this case, compromise behaviors (see Section 4.2.6) are possible and then several motivations can be satisfied at the same time.

With this pro-active level, the action selection model can manage conflicting goals generated by a motivational level according to the environmental perceptions. The behavioral planner determines the sequence of intermediate actions necessary for executing the final action that will satisfy the motivation.

4.1.3 Motivational level

Motivations are the principal weight when making a decision and the drive behaviors most of the time (see Section 2.5.1). The activities of motivations are propagated inside the system and mixed with the environment information (see Figure 4.5). Then, action selection mechanisms have to find the most appropriate action among all the possible ones.

Motivation activities come from internal needs and have to be distinct because internal needs evolved differently and the perceptions are different for each entity. That is why we design our model of action selection for one virtual human first: to well understand the implications of individuality in the decision making. A decision making system has to be individual in order to obtain more interesting and believable behaviors even in social environments. The social interaction is based on the difference between the entities and most of the time they are too similar and behave identically.

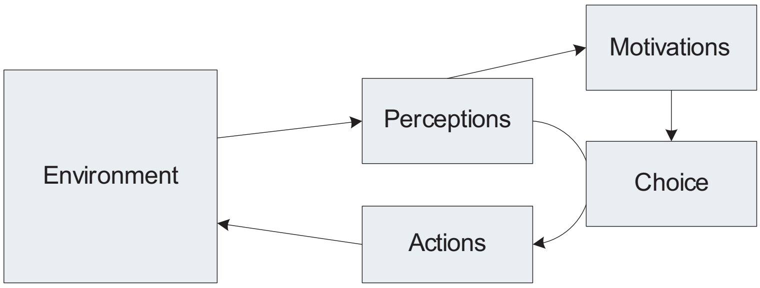

Figure 4.5: Schematic view of the motivational level

In our model, three levels of motivations are designed, inspired from Maslow's pyramid of needs [Maslow 54] (see Section 2.5.1). The **basic motivations** such as hunger, thirst, or rest have the priority over the other motivations. It is comparable to Brook's subsumption architecture [Brooks 86] for the priorities between levels. The second level represents **essential motivations** such as clean, sleep, or wash. Finally the third level corresponds to the **secondary motivations** such as read, watch TV, or water plants. When all the basic motivations are satisfied, the virtual human can consider the essential ones and the same for the secondary ones.

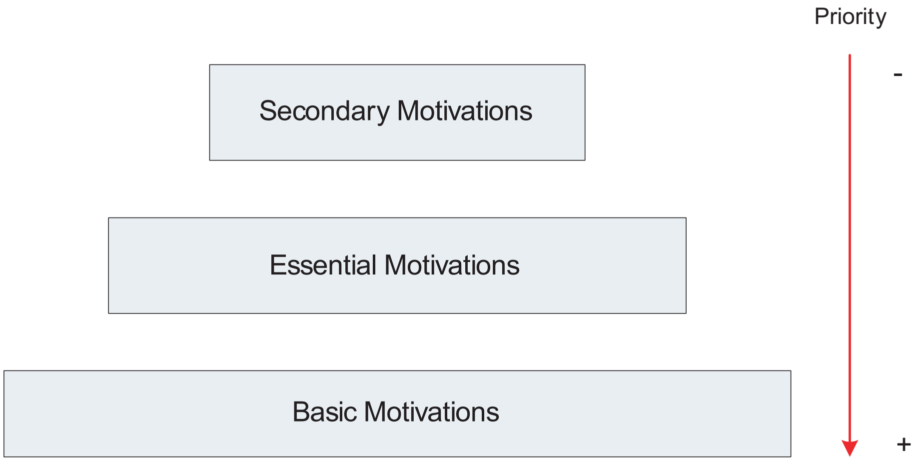

Figure 4.6: Hierarchy of motivations inspired from Maslow's hierarchy of needs

This corresponds to a survival criterion studied in ethology and it is also linked with the evolutionary process. Primitive animals have only basic motivations whereas complex animals can have secondary motivations. Hunger or thirst are more important for animals and humans than reading or playing. Therefore, if a basic motivation and an essential motivation have the same high value, the basic one has to be chosen. It is the same between essential motivations and secondary ones. It helps the decision making system to choose in real-time the most appropriate action according to the motivations and the environmental information.

However, this layered solution doesn't solve entirely the oscillation avoidance problem during the choice between several goals (see Section 2.1.3). So, a threshold system has been implemented to focus the attention of the virtual human on the most urgent motivation to satisfy. In our system, it corresponds to a "subjective" evaluation of the motivations compared to the internal needs (see Section 4.2.3). This system is composed of two thresholds delimiting three zones: the comfort zone, the tolerance zone and the viability zone inspired from [Ashby 52, Meyer 95, Cañamero 97]. The role of the action selection mechanism is to maintain the internal needs (the origins of the motivations) inside the comfort zone like a homeostasis system. On the other hand, if internal needs are situated inside the viability zone, the value of the corresponding motivation is amplified accordingly in order to satisfy rapidly this motivation. It can be assimilated with attention mechanisms and in other models, emotions are playing this role.

In order to return the internal needs inside the comfort zone, instead of decreasing the motivations, a hysteresis (see Section 4.2.4) has been implemented to maintain a part of the previous value of the motivations in the calculus of the current one. If the current motivation decreases because the corresponding action satisfies it, it will be no longer the highest motivation and then the system will switch to another motivation. However, with the hysteresis, the value of the motivation which is satisfied is a composition between the current value and the value of the previous iteration. In this case, the current motivation is maintained high and then the internal needs have time to decrease inside the comfort zone.

Hysteresis helps also to define the duration of actions according to the context of the simulation, the motivations, and the environmental information. Some motivations are periodic like mealtimes or living work. Others are contextual and occasional, like phoning or drinking. The difficulty is to know whether the motivations should be based on a twenty four hour day like real humans, and how long should each action last. Indeed, the sleeping action should last at least a few minutes; otherwise it is a resting action. All the actions have a normal duration that we define arbitrarily and this duration changes a little to fit with the context of the simulation.

Finally, we claim that without a motivational level implemented in the decision making model, the virtual humans (or other virtual characters such as animals and even robots) cannot be really autonomous or have an individuality before focusing on emotions, cognition or social interactions (see Section 2.5.1). As motivations allow to self-generate goals, specific to each virtual human, they give the illusion of living their own life increasing the believability of persistent virtual environments. However, the user keeps the control by defining all the parameters in the graphical interface, or by scripting some behaviors (see Section 4.3.4). A motivation is defined by default according to the scenario and the environment. For instance, in an office, the default motivation is normally to work. Then, other motivations such as hunger, thirst, phoning, living, etc. evolve in parallel and the system has to choose the most appropriate action according to the motivations and the environmental information.

4.1.4 Emotional level

The next step after designing the motivational model of action selection is to add an emotional level in order to increase the flexibility and sociability of the model. Emotions enhance also the autonomy and the individuality of the virtual humans by giving them a certain personality (see Section 2.5.2). The contributions of the emotions compared to the motivations are summarized by Canamero [Cañamero 97] in Table 4.1.

Problems with reactive and motivated architectures	**Contributions of emotions**
Rigidity of behaviors (S / R)	More flexible and varied behavior as a function of internal state (e.g. predator: attack or flee)
Insufficient autonomy (reactions, drives)	Modulation/change of motivational state and behavior
Repetitive and inefficient behaviors (loops)	Self-monitoring, interruption of inefficient behavior
Inefficient treatment of urgency situations, "goal forgetfulness"	faster responses, anticipation (emotional memory) Re-equilibration of internal milieu, back to goal
"Atomic" behavior	Behavioral chains (e.g. fear, escape, anger, attack, relief)

Table 4.1: Contributions of the emotions compared to the motivations [Cañamero 97]

The three main advantages of emotions in action selection are [Cañamero 01]:

- Have rapid reactions (fast adaptation)
- Contribute to resolve the choice among multiple goals
- Signal relevant events to others

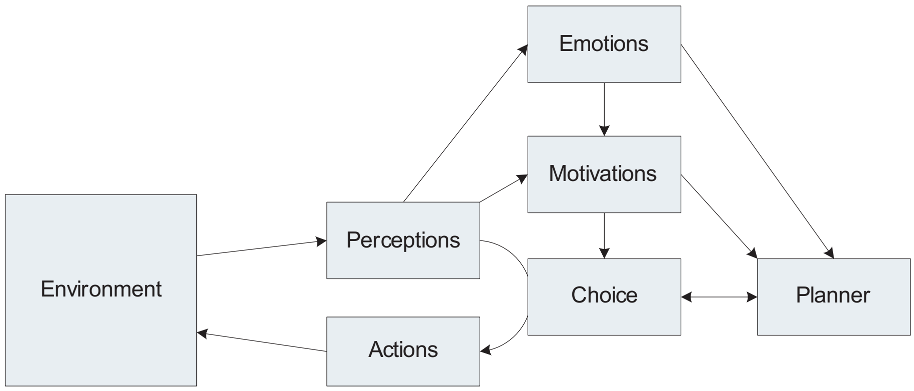

Figure 4.7: Schematic view of the emotional level

The principal problem when you implement an emotional level in a motivational decision making mechanism is to know which emotions will influence which parameters in the model. The role of emotions is less clear than the motivation one. On the one hand, since an emotional system is a complex system connected to many other behavioral and cognitive subsystems, it can act on all these other systems at the same time. Their effects are more subtle and they evaluate and modulate many things. In our model, they can modulate several parameters such as the threshold of the subjective evaluation for each motivation or the activation threshold, the length, and the effects on internal needs for each action. They have also an influence on hysteresis for each motivation and on perception for specific objects or in a general way.

On the other hand, since emotions are related with goals, rather than with particular behavioral response patterns [Frijda 95, Rolls 99], they contribute to the generation of richer, more varied, and flexible behaviors [Cañamero 01]. The emotional models in psychology are often too complex to be implemented in a decision making mechanism. Motivations are mostly distinctive and they have to be satisfied rapidly whereas the emotions influence the decisions over the long term. Moreover, the emotions are linked with the external context and with social interactions. For example, when the virtual human is sad, he cannot "satisfy" his sadness quickly whereas if he is hungry, he just has to eat. Emotions are more comparable to states that influence all the decisions on the long term, except for urgent situations. In fact there are many degrees of emotional controls depending on the external situations.

In our model, many roles that normally are attributed to emotions (see Table 4.1) are already managed without emotions principally thanks to the subjective evaluation of the motivations (see Section 4.2.3). Specific motivations can take the control with a high priority in order to protect the virtual humans (see Section 4.3.1). It is the same in the nature when an animals escape from predators for their survival. The subjective evaluation of motivations also allows to stop actions when the internal needs are back in the comfort zone. It is possible to define interactions between the motivations in order to chain them (see Section 4.3.2) and to interrupt the current behavior if another motivation is more urgent to satisfy or an opportunist behavior occurs (see Section 4.2.7).

Emotions evaluate and influence decision making and manage social interactions. For more autonomy, individuality and realism of the virtual humans, we begin to add an emotional level in our model (see Section 4.3.3).

4.2 The motivational action selection

Motivations constitute urges to action based on internal needs related with survival and self-sufficiency. As they drive behaviors in normal situations, they are the core of the model, decision-making grouping the three first levels. Our motivational action selection model fulfills Tyrrell's criteria (see Section 3.2.6) for implementing effective action selection mechanisms and the requirements for designing autonomous virtual agents (see Section 2.4.2). The emotional level is at the beginning of its integration in the decision-making process following the bottom-up approach (see Section 2.1.1) in order to integrate social interactions in the model.

4.2.1 Model description

The model of action selection is composed of overlapping hierarchical decision loops running in parallel. The number of motivations is not limited. Hierarchical classifier systems contain the three following levels: motivations, goal-oriented behavior, and actions. The activity is propagated throughout the hierarchical classifier systems according to the two rule conditions: internal context and environment perceptions. Selection of the most activated node is not carried out at each layer, as in a classical hierarchy, but only in the end in the action layer, as in a free flow hierarchy. In the end, the chosen action is the most activated one.

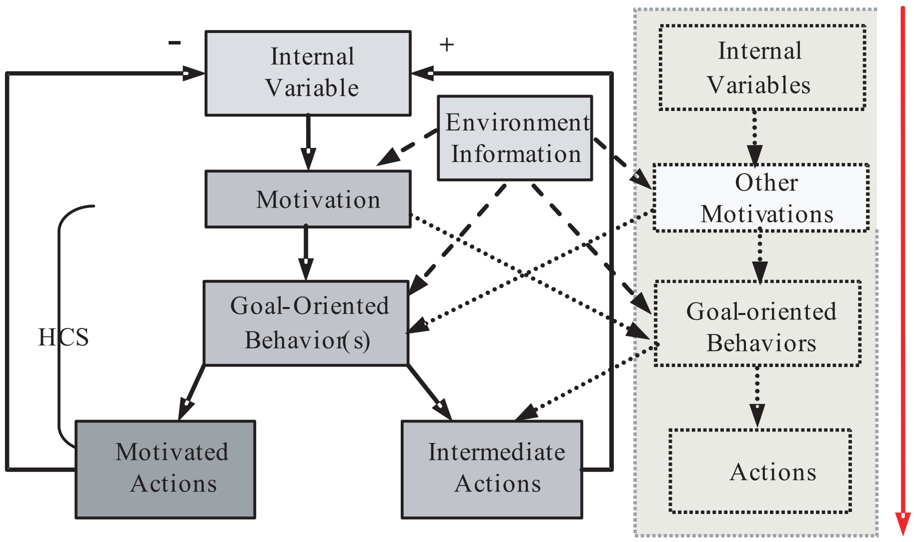

Figure 4.8: A hierarchical decision loop of the model for one motivation (activity at each iteration). HCS corresponds to the hierarchical classifier system

In the model, hierarchical decision loops, one per motivation, are running in parallel. For clarity purposes, Figure 4.8 depicts a hierarchical decision loop for one motivation. It contains four layers:

1. **An internal variable** represents the homeostatic internal state of the virtual human and evolves according to the effects of actions. The action selection mechanism should maintain the internal variables within the comfort zone.

2. **Motivation** is an abstraction corresponding to the tendency to behave in particular ways according to the environmental information, a "subjective evaluation" of the internal variables, and a hysteresis. Motivations set goals for the virtual human in order to satisfy internal variables.

3. **The Goal-oriented behaviors** represent the internal context of the hierarchical classifier system. They are used to plan sequences of actions such as reaching specific goals. In the end, the virtual human can perform motivated actions satisfying the motivations.

4. **The Actions** are separated into two types. *Intermediate actions* are used to prepare the virtual human to perform *motivated actions* that can satisfy one or several motivations. Intermediate actions often correspond with moving the virtual human to specific locations in order to perform motivated actions. Both have a retro-action on internal variables. Intermediate actions increase them, whereas motivated actions decrease them.

An example of the hierarchical decision loop for the motivation "hunger" depicted in Figure /refmodelEx helps to understand how the model of action selection for one motivation works.

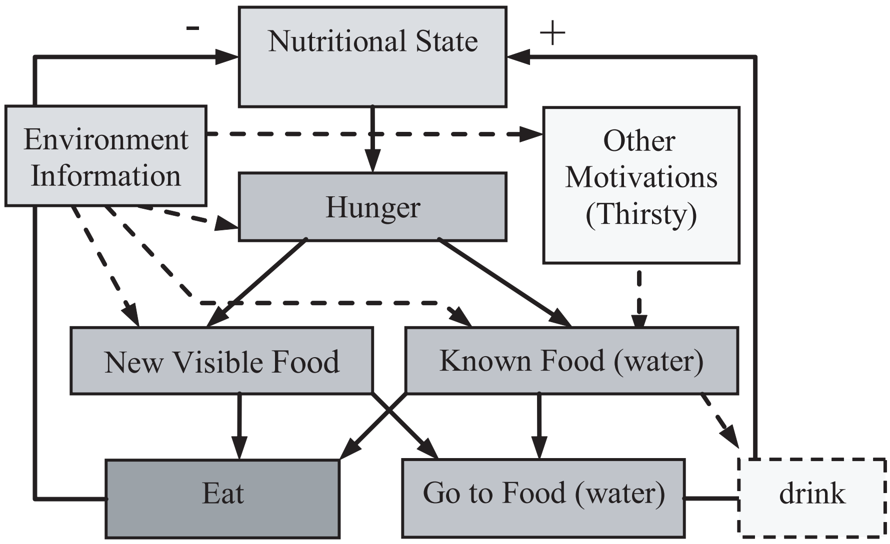

Figure 4.9: Hierarchical decision loop example: “hunger”

4.2.2 Environmental information and opportunistic behaviors

To satisfy his motivations autonomously, the virtual human should be situated in his environment, i.e., he should sense his environment through his sensors and act upon it using his actuators [Maes 94] (see Section 2.4.2). Therefore, the virtual human has a limited perception system which can sense his environment. He can then navigate in the environment, avoiding obstacles with the help of a path-planner, and satisfy motivations by moving to specific goals and interacting with objects. The environment perception is integrated in the model at two levels: in motivations and in goal-oriented behaviors.

Indeed, two types of opportunist behaviors, which are the consequences of the reactivity and the flexibility of the model, are possible. First, when the virtual human passes near a specific place where a motivation can be satisfied, the value of this motivation (and the respective goal-oriented behaviors) is increased proportionally to the distance to this location and inversely. For example, when someone passes near a cake and sees it, even if he is not really hungry, his level of hunger increases. Second, when the virtual human sees on his way a new, closer location where he can satisfy his current motivation, the most appropriate behavior is to interrupt dynamically his current behavior to reach this new location, instead of going to the original one.

4.2.3 "Subjective evaluation" of motivations

Instead of designing a winner-take-all hierarchy with the focus on attention [Bryson 00, Blumberg 96], we develop a "subjective evaluation" of motivations corresponding to a non-linear model of motivation evolution. It allows to have a selective attention with the advantages of free-flow hierarchies: unrestricted flow of information, possibilities of compromise and opportunist candidates (see Section 3.2.4). A threshold system, specific to each motivation and inspired by the viability zone concept [Ashby 52, Meyer 96], reduces or enhances the motivation values to maintain the homeostasis of the internal variables. One of the main roles of the action selection mechanism is to preserve the internal variables within the comfort zone by choosing the most appropriate actions. This threshold system can be assimilated with degrees of attention. It limits and selects information to reduce the complexity of the decision making task [Bryson 00]. In other models, emotions could play this role [Cañamero 97]. This system also helps to make a choice among multiple and conflicting goals at any moment in time and reduces the chances of dithering or pursuing a single goal to the detriment of all others.

We model the subjective evaluation of motivations as follow:

$$\begin{cases} M = T_1 e^{(i-T_1)^2} & \text{if } i < T_1 \\ M = i & \text{if } T_1 \leq i \geq T_2 \\ M = \frac{i}{(1-i)^2} & \text{if } i > T_2 \end{cases}$$

where M is the motivation value, T_1 the first threshold, T_2 the second threshold and i the internal variable.

If the internal variable i lies beneath the threshold T_1 (comfort zone), the virtual human does not pay attention to the motivation. If i is between both thresholds (tolerance

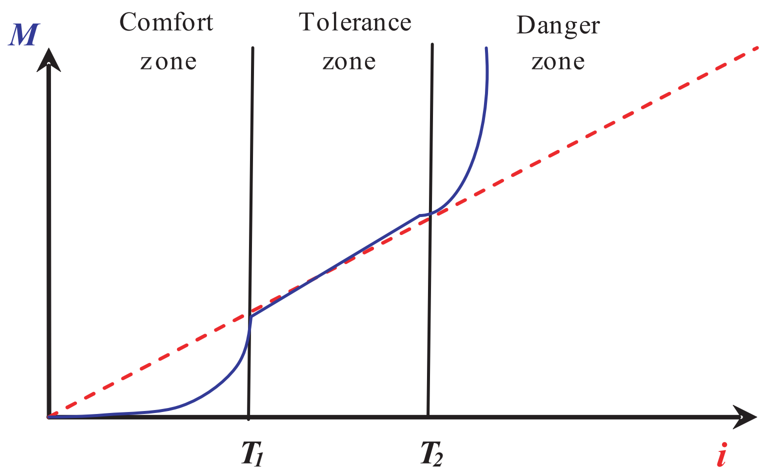

Figure 4.10: "Subjective" evaluation of one motivation from the values of the internal variable.

zone), the value of the motivation M equals the value of the internal variable. Finally, if i is beyond the second threshold T_2 (danger zone), the value of the motivation is amplified in comparison with the internal variable. In this case, the corresponding action has more chances to be chosen by the action selection mechanism, to decrease the internal variable.

4.2.4 Hysteresis and persistence of actions

The inhibition system plays an important role in ethological models of action-selection and is used to explain some of the temporal aspects of behavior. Yet animals typically do not mindlessly pursue an activity indefinitely to the detriment of other needs (see Section 2.1.3). Indeed, animals sometimes appear to engage in a form of time-sharing [McFarland 74, McFarland 93]. Blumberg [Blumberg 94] implements an inhibition and fatigue system for avoiding behavioral dithering (the rapid oscillation between different behaviors) and incorporates time-sharing for low priority behaviors to execute in the presence of high priority behaviors. While animals typically do not dither between multiple activities, they will nonetheless interrupt a behavior when another behavior becomes significantly more appropriate.

The difficulty is to control the temporal aspects of behaviors so as to arrive at the right balance between too little persistence, resulting in dithering among activities, and too much persistence so that opportunities are missed or that the agent mindlessly pursues a given goal to the detriment of other goals [Blumberg 96]. Instead of using an inhibition and fatigue algorithm where the difficulty resides in defining inhibitions between motivations (for example watering and eating), a hysteresis has been implemented, specific to each motivation, to keep at each step a portion of the motivation from the previous iteration. In addition to the "subjective evaluation" of the motivation, the hysteresis allows the persistence of motivated actions and the control of temporal aspect of behaviors.

We model the hysteresis as follow:

$$M_t = (1 - \alpha) \cdot M_{t-1} + \alpha(M + e_t)$$

where M_t is the motivation value at the current time, M the "subjective" evaluation of the motivation, e_t the environment variable and α the hysteresis value with $0 \leq \alpha \leq 1$.

The goal is to maintain the activity of the motivations and the corresponding motivated actions for a while, even though the value of the internal variable decreases. Indeed, the chosen action must remain the most activated one until the internal variables have returned within their comfort zone. Someone goes on eating even if the initial feeling of hunger has disappeared, and stops only when he has eaten his fill. This way, the hysteresis limits the risk of action selection oscillations between motivations.

4.2.5 Behavioral sequences of actions

To satisfy the motivations of the virtual human by performing motivated actions, behavioral sequences of intermediate actions need to be generated (see Section 3.1.5), according to environmental information and the internal context of the hierarchical classifier system [Donnart 94].

In Table 4.2 and Figure 4.11, hunger is the highest motivation and must remain so until the nutritional state is back within the comfort zone. The behavioral sequence of actions for eating needs two internal classifiers (R_0 and R_1) and three external classifiers (R_2, R_3 and R_4):

R_0: if "known food location" and "the nutritional state is high", then "hunger".

R_1: if "known food is remote" and "hunger", then "reach food location".

R_2: if "reach food location" and "known food is remote", then "go to food".

R_3: if "near food" and "reach food location", then "take food".

R_4: if "food near mouth" and "hunger", then "eat".

Here, the virtual human should do a specific and coherent sequence of intermediate actions in order to eat and then satisfy his hunger. In this case, two internal messages "hunger" and "reach food location" are added to the message list, thanks to the internal

Time steps	t_0	t_1	t_2	t_3	t_4	t_5	t_6
Environmental information	known food location, but remote				Near food	Food near mouth	No food
Internal context (Message List)		hunger					
			reach food location				
Actions				Go to food	Take food	Eat	
Activated rules		R_0	R_1	R_2	R_3	R_4	

Table 4.2: Example for generating a sequence of actions using a hierarchical classifier system (timeline view).

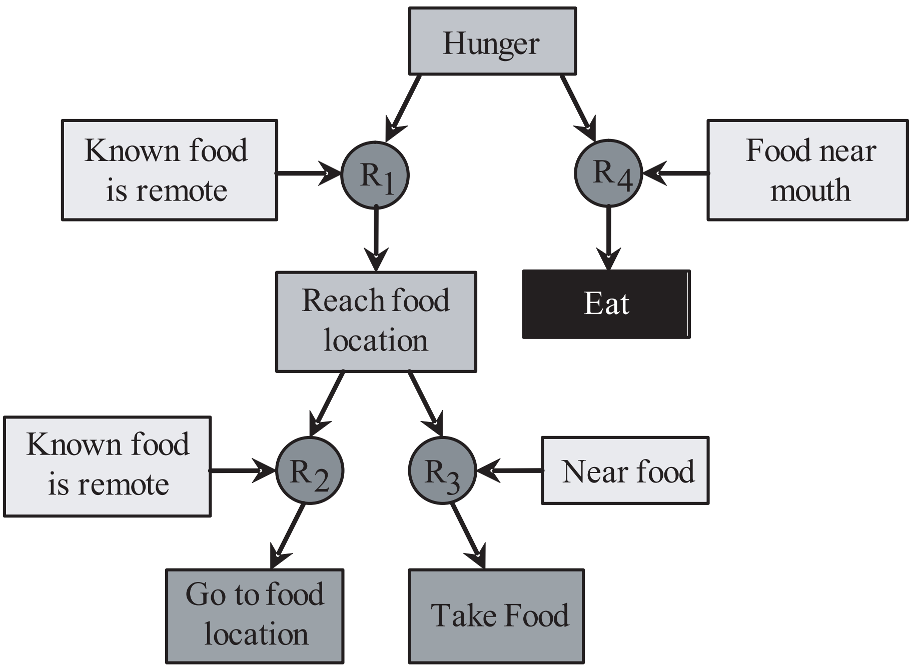

Figure 4.11: Example for generating a sequence of actions using a hierarchical classifier system (hierarchical view).

classifiers R_0, and then R_1. They represent the internal state for the rules and remain until they are performed. To reach the known food location, two external classifiers, R_2 and R_3, activate intermediate actions as many times as necessary. When the virtual human is near the food, the internal message "reach food location" is deleted from the message list and the last external classifier R_4 activates the motivated action "eat", decreasing the nutritional state. Furthermore, motivated actions weigh twice as much as intermediate actions because they can satisfy motivations. They therefore have more chances to be chosen by the action selection mechanism. Thereafter the internal message "hunger" is deleted from the message list, the food has been eaten and the nutritional state has returned within the comfort zone for a while.

Hierarchical classifier systems allow to chain actions in order to have coherent behaviors and to satisfy motivations at specific locations.

4.2.6 Compromise behaviors

In our model, the activity coming from the motivations is propagated throughout the hierarchical classifier systems, according to the free flow hierarchy [Tyrrell 93a]. A greater flexibility in the behavior, such as compromise behaviors (see Section 3.2.3) where several motivations can be satisfied at the same time, is then possible. The compromise behaviors have also more chances of being chosen by the action selection mechanism, since they can group activities coming from several motivations.

We model the compromise behaviors as follow:

$$A_c = A_h + \frac{\sum_m^1 \beta A_i}{\sum_n^1 M_i} \cdot A_h \quad \text{if } A_m \geq S_c$$

where A_c is the compromise action activity, A_h the highest activity of compromise behaviors, β the compromise factor, A_i the compromise behaviors activities, m the number of compromise behaviors, M_i the motivations, n the number of motivations, A_m the lowest activity of compromise behaviors and S_c the activation threshold specific to each compromise action.

Compromise actions are activated only if the lowest activity of the compromise behaviors is over a threshold defined within the rule of the compromise action. The value of the compromise action should always be based on the highest compromise behavior activity even if it is not the same one until the end. The corresponding internal variables should return to the comfort zone. However, the other internal variables concerned by the effect of the compromise action stop to decrease from an inhibition threshold to keep the internal variables standardized. For instance, let's say the highest motivation of a virtual human is "hunger". If he is also thirsty and knows a location where there are both food and water, even if it is farther, he should go there. Indeed, this would allow him to satisfy both motivations at the same time, and both corresponding internal variables would return within their comfort zone.

4.2.7 Behavior interruptions

In the opposite of most of the traditional behavior planners in which a plan should be finished before controlling the environment changes, our planner can be interrupted at any time by another more urgent behavior. According to Hawes [Hawes 01], behavior interruptions are very important for flexible behaviors in real-time and dynamic environments especially for unexpected situations such as danger or opportunist behaviors. Most of the behavioral planners cannot be interrupted. It prevents the virtual human from continuing a plan that is not optimal any more because of the environment constant changes. In this case, the virtual human's choice of actions fits continuously with the environment variations. In case of fire, the virtual human has to stop the current plan rapidly and run away or put out the fire. Other possible interruptions are opportunist behaviors. They are indispensable for environmental adaptation of the behaviors. It is one of Tyrrell's criteria for designing efficient action selection mechanisms [Tyrrell 92]. If the virtual human is hungry and on his way he perceives another source of food, he should stop his behavior and eat at the new source instead of continuing his plans. The system has to be situated and has to choose the most appropriate action in order to have coherent but flexible behaviors.

4.3 Influences on motivational decision-making

Several functionalities are implemented to help and evaluate the decision of the motivational action selection model. They allow to have coherent but flexible behaviors for autonomous virtual humans in persistent worlds.

4.3.1 Managing dynamic and unexpected environment

Action selection mechanisms should manage dynamic and unexpected environments in real time [Guillot 98, Cañamero 01]. Then, the user can define subjective or physical quantities in the model rules at the initialization and modify them during the simulation.

For the food or water sources, physical quantity can be set with the initial and the maximum value and also the variation factor. On the other hand, for the source where the virtual human can phone or clean, it is more subjective quantities that can be defined corresponding to the time left until the phone will break down or the sponge will be used.

Dynamic environments also imply unexpected situations and even dangerous ones. The virtual human has to adapt his behavior to these situations and the action mechanism should select the appropriate action for the virtual human survival. In many models [Cañamero 98, Rank 05], emotions were responsible for evaluating and adapting behaviors whereas, in our case, the subjective evaluation of motivations plays this role. If the virtual human sees a danger for his life (for example "fire"), the motivation that is the best for his survival will be the highest priority ("escaping the danger") and the corresponding action ("move away") will be chosen until the danger disappears (see Section 7.2.6). This action is always the highest action:

$$\begin{cases} A_a = \alpha & \text{if all the } A_i < \alpha \\ A_a = A_h + \beta & \text{if } A_h >= \alpha \end{cases}$$

Where A_a is the activity of the avoid action, α is the danger threshold, A_i the activities of action i, A_h the highest action if the danger threshold is overcome and β is the danger factor.

The danger threshold and factor have some default values but can be defined by the user. In the end, the behavior of the virtual human is more adaptive and realistic thanks to this management of dynamic and unexpected environments by the system.

4.3.2 Behavior interactions

For chaining behaviors and not only actions, interactions can be defined in the comportment rules and modified during the simulation. The users can set several logic interactions between motivations with their influence factors. For instance, if the virtual human eats or drinks, it increases his desire to go to the toilets. Likewise if the virtual human does some push-ups, he should take a bath and have some rest. Moreover, the influence factor allows to set the strength of the interaction. The highest the influence factor is, the more the action implied in the interaction has chances to be chosen by the action selection mechanism. However, some opportunistic behaviors, danger situations, or more urgent motivations to satisfy can take the control instead. For instance, if the virtual human cooks, the influence factor can be set so that the action selection mechanism chooses preferentially the "eat" action once the "cook" action is performed (see section 7.3.4). It corresponds to a preference for chaining behaviors instead of rigid links that must be performed absolutely. In the end, with these behavioral interactions, it is possible to have complex sequence of behaviors while keeping the flexibility of the system.

4.3.3 Emotion influences

The roles of emotions are not so clear than those of the motivations. Their influences are multiple and act at several levels (see Table 4.1). In our model, the subjective evaluation of the motivations can play many roles attributed to emotions (see section 4.1.4). Therefore we just begin to implement the emotional level. It is developed principally to enhance the autonomy of the virtual humans and in the future to manage social interactions. But in a "bottom-up" approach, one begins with simple functionalities to understand how it works and then complexifies it progressively. For now, only the like/dislike notion is managed and has an influence on the activation thresholds of the actions. The emotional influences have a weight in the decision making, because the users have defined whether the virtual human likes or dislikes each action. This is defined randomly at the beginning of the simulation if nothing is set in the initialization file, but can be modified during the simulation thanks to the graphical interface. If the virtual human likes the action, the action selection mechanism will choose this action preferentially and vice versa for the disliked actions. It enhances the individuation and personality capabilities of the virtual humans in order to be more realistic and believable because the behaviors are less rigid and fixed.

4.3.4 External control

In spite of the decision making system that gives the virtual humans autonomy, the users can take control of them with simple commands written in the VHD++ python editor [Schertenleib 05]. A specific scenario can be followed according to a script, as for storytelling, but it is not the aim of this work. Python [Grayson 00, Lutz 96] is very powerful but a little more complex than behavioral specific programming languages. Some macro-commands exist in VHD++ (see Section 5.1) to access general functions such as loading and positioning virtual characters and objects, walking to a specific location avoiding obstacles, playing keyframes, etc. However, our ultimate goal is that the user doesn't have to use it, which implies that the virtual humans are autonomous, adaptive, and realistic.

4.4 The choice of actions

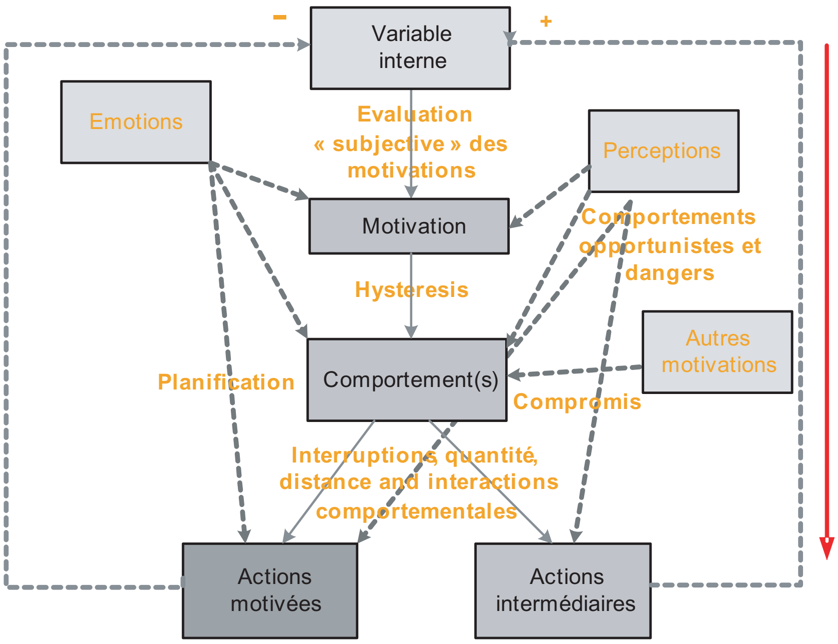

Figure 4.12: All the factors influencing the activity propagation

As the activity is propagated throughout the hierarchy, and the choice is only made at the level of the actions, the most activated action is the most appropriate one at each iteration. This depends on the "subjective evaluation" of the motivations, the hysteresis, the environmental information, the internal context of the hierarchical classifier system, the weight of the classifiers and the other motivation activities, the behaviors interactions and the emotional influences. Most of the time, the action receiving activity coming from the highest internal variable is the most activated one, and is thus chosen by the action selection mechanism. Normally, the corresponding motivation stays the highest one until the current internal variable decreases in the comfort zone. Behavioral sequences of actions can then be generated by choosing intermediate actions at each iteration. Finally, the motivated action required to decrease the internal variables is performed. In the case of choice conflicts, i.e., actions with almost the same values, the goal distance is taken into account and the model chooses the nearest one. However, other factors can influence the decision-making, e.g., the quantity of the source if any and the possibility of compromise behaviors. For instance, if the virtual human can perform a compromise action even if it is farther (within a certain range), he will choose it preferentially instead of following sequentially two motivated actions. On the other hand, if the quantity of the source is very low compared to the others, he will not choose the compromise action. The influence of these three factors on the choice is a pondered value according to the priority. However, there are exceptions, e.g., when another motivation becomes more urgent to satisfy, or when opportunist behaviors, as well as dangers, occur. In these cases, the current behavior is interrupted and a new behavioral sequence of intermediate actions is generated so that the virtual human satisfies this new motivation.

4.5 Summary

Our multi-layered action selection architecture respects the requirements (see Section 2.6) that we have summarized for designing effective action selection mechanisms for autonomous virtual humans in persistent worlds because of its four levels:

- reactive: managing unexpected and opportunist situations,
- pro-active: allowing to satisfy motivations by generating goal-oriented behaviors to reach specific goals,
- motivational: responsible for the virtual human autonomy by giving the goals to the system,
- emotional: modulating and evaluating the system choices and necessary for social interactions.

We focus on the first three levels in a bottom-up approach (see Section 2.1.1). The emotional model is at the beginning of its implementation and we try to understand how the emotions influence the other levels. The decision-making of the motivational action selection model is flexible but coherent in order to satisfy self-generated motivations at specific locations according to the environmental perceptions. Virtual humans have then a strong autonomy (see Section 2.5.1) and can be able to live their own lives in persistent worlds. The additional influences allow to have less rigid and pre-determined behaviors. Emotions enhance the autonomy and the individuation of the virtual humans by giving them a certain personality. Finally the choice of actions depends on many influences and becomes difficult to predict, thus providing more interesting and believable behaviors for the autonomous virtual humans in persistent worlds.

Chapter 5

Model Implementation

The model of action selection was implemented in Python, which is a script language. Therefore it is useful to test some functionalities of the model without recompiling all the code. The implementation was done so that the users can configure all the parameters easily and the system manages it automatically. For instance, the keyframes should be put in a specific directory. Then, they are created, loaded and viewed thanks to the graphical interface. The rules are also simple to define and to change and the decision making system takes them directly into account. The implementation was also done so that the users can set the parameters as they want, because there is no general tuning for these parameters. We have defined default values for them in accordance with what we think but we don't claim that it is the good tuning. So if the user doesn't agree, he can change the parameters in the initialization file and with the graphical interface to make them match with his ideas. In this kind of simulation, we claim that learning and automatic tuning are not possible. That is why we opt for a full configurable simulation. Moreover, the system is not specific to one virtual human or to one environment and even to computer graphic. The model can even be run without graphic output. The number of rules is not limited and they are highly configurable by the users to match with their experiences.

5.1 Platform for character simulation

The VHD++ [Ponder 03] component-based framework engine developed by VRLAB-EPFL and MIRALab-UNIGE is depicted in Figure 5.1. It allows quick prototyping of virtual reality - augmented reality applications featuring integrated real-time virtual character simulation technologies.

Component-based framework system design

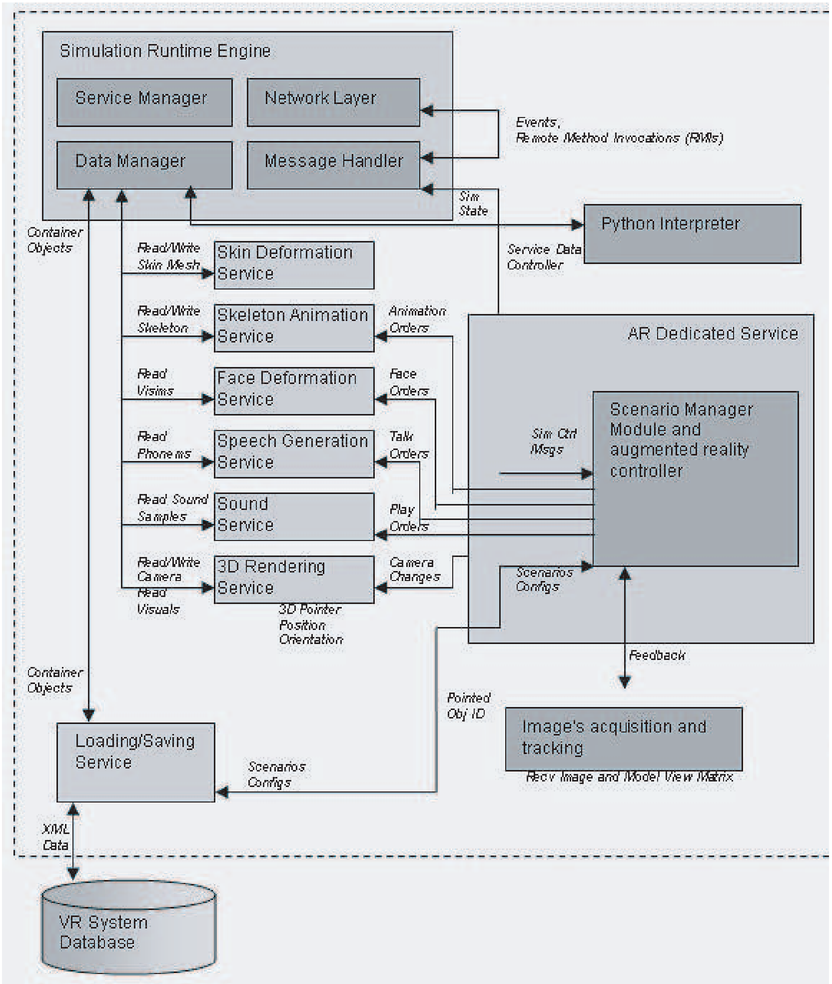

Figure 5.1: VHD++ AR framework overview

The framework has borrowed extensive know-how from previous platforms such as presented by Sannier et al [Sannier 99]. The key innovation is focused on the area of component-based framework that allows the plug-and-play of different heterogeneous human simulation technologies such as: Real-time character rendering in augmented reality (supporting real-virtual occlusions), real-time camera tracking, facial simulation and speech, body animation with skinning, 3D sound, cloth simulation and behavioral scripting of actions. The integration of the augmented reality framework tracking component is based on a two-stage approach. Firstly the system uses a recorded sequence of the operating environment in order to train the recognition module. The recognition module contains a database with invariant feature descriptors for the entire scene. The runtime module then recognizes features in scenes by comparing them to entries in its scene database. Secondly, by combining many of these recognized features it calculates the location of the camera and thus the user position and orientation in the operating environment. The main design principle was to maximize the flexibility while keeping excellent real-time performance. The different components of the framework may be grouped into the two following main categories:

- System kernel components responsible for the interactive real-time simulation initialization and execution.
- Interaction components driving external VR devices and providing various GUIs allowing interactive scenario authoring, triggering, and control.

Finally, the content to be created and used by the system is specified: it may be classified into the two following main categories: a) Static and b) Dynamic content building blocks such as models of the 3D scenes, virtual humans, objects, animations, behaviors, speech, sounds, python scripts, etc.

Framework operation for character simulation

The software architecture is composed of multiple software components called services, as their responsibilities are clearly defined. They have to take care of rendering 3D simulation scenes and sound, processing inputs from the external VR devices, animating the 3D models and in particular complex animations of virtual human models including skeleton animation and respective skin and cloth deformation. These services are also responsible for maintenance of the consistent simulation and interactive scenario state that can be modified with python scripts at run-time. To keep good performance, the system uses four threads. One thread is used to manage the updates of all the services that we need to compute, such as human animation, cloth simulation or voice (sound) management. A second thread is used for the 3D renderer (see Figure 5.2), which obtains information from the current scenegraph about the objects that must be drawn as well as the image received from the camera. It changes the model view matrix accordingly to the value provided by the tracker.

The third thread has the responsibility of capturing and tracking images. The last thread is the python interpreter [Schertenleib 05] (see Figure 5.3), which allows us to create scripts for manipulating our application at the system level, e.g., generating behaviors

Figure 5.2: View of the 3D viewer module of VHD++.

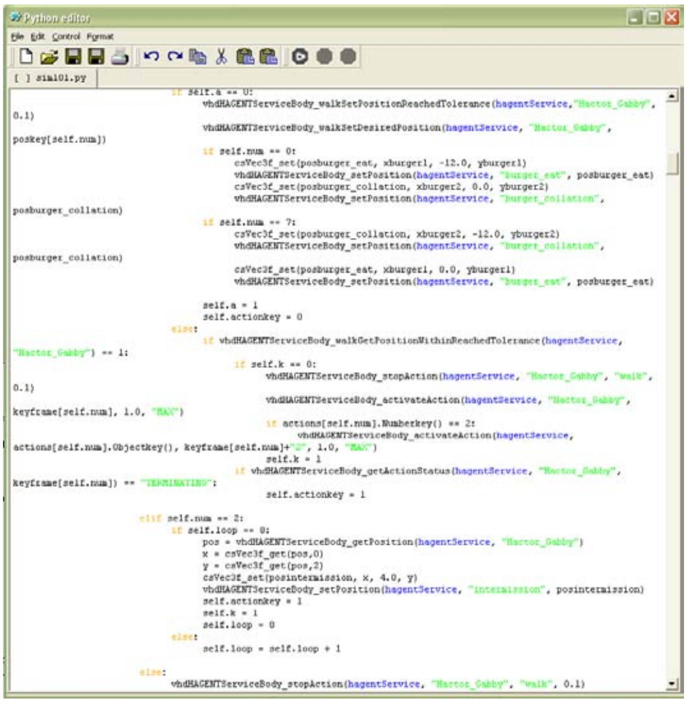

Figure 5.3: View of the python editor module.

for the human actions (key-frame animation, voice, navigation). Through the python editor, we can access and manage all the functions of VHD++ quoted above, such as viewing a keyframe in the 3D viewer, interacting with objects, manipulating cameras...

Supporting the choice of a multi-threaded application model came first from the consideration of keeping smooth human animation frame rate as well as using technological features like Hyper-Threading [Technology 04].

The augmented reality system presented in Figure 5.1 features immersive real-time interactive simulation supplied continuously with proper information. That is why content components are much diversified and thus their development is an extremely laborious process involving long and complex data processing pipelines, multiple recording technologies, various design tools and custom made software. The various 3D models to be included in the virtual environments like virtual humans or auxiliary objects have to be created manually by 3D designers. The creation of virtual humans requires to record motion captured data for realistic skeletal animations as well as a database of real gestures for facial animations. Sound environments, including voice acting, need to be recorded in advance based on the story-board. For each particular scenario, dedicated system configuration data, the physical environment and the VR device parameters have to be defined as well as scripts defining atomic behaviors of simulation elements. These scripts can modify any data used by the current simulation in real-time. This allows us to continue running the simulation whilst some modifications are performed.

Data driven programming design - the development of a VR scripting engine

Many VR systems continue to rely on monolith architecture due to the simplicity in the coding implementation but also to prevent that external people hijack their resources. However, today virtual environment complexity needs to be subdivided so that they remain manageable. Using a monolithic application will require to recompile the application every time a designer needs to modify a parameter. Thus, the programmers become a bottleneck within the development process. By separating into components, we allow a more synergistic move, as each of these aspects can prosper independently from the other. One of the key ideas with a data-driven architecture is to keep the core functionalities as minimal as possible for both performance reasons and giving more control to designers. This allows to prototype complete discrete applications using the same core of functionalities without extending the engine with specific cases that will be deprecated over time and prone to errors.

Threading model

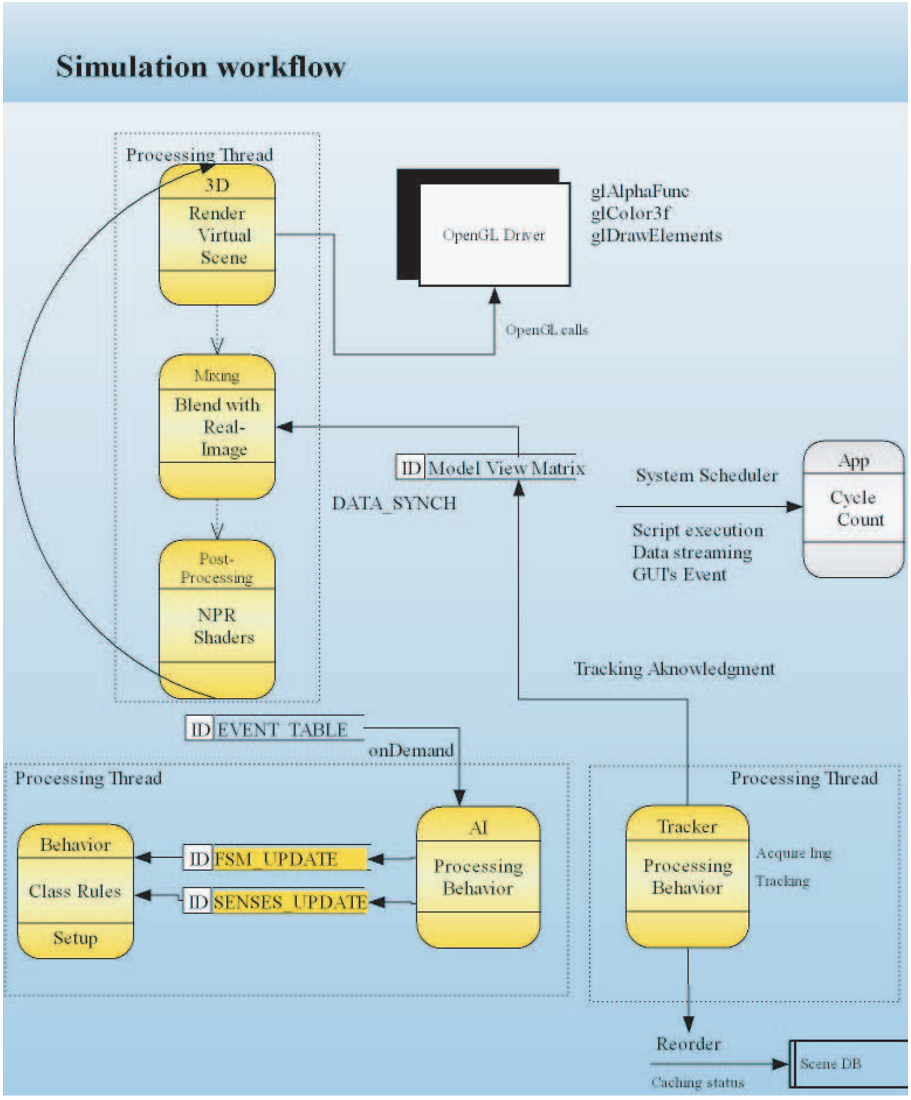

Figure 5.4: Threading model

Our approach is consistent and tends to separate the individual flow of information between threads, as described in Figure 5.4. Every process has to be as independent as possible for reducing bottlenecks on data synchronization. For instance, we clearly separate the 3D rendering from the artificial intelligence and the simulation events from script management. By separating different components within our architecture, we are able to generate more complex simulations with the same level of interactivity.

5.2 Rule based system

All rules are defined in an initialization file and can be modified during the simulation. The user can edit easily this file to modify the existing rules or to add/delete some. A syntax using Python lists has to be respected for each level of the hierarchy (see the Appendix A for example):

Internal Variable

[name, type, initial value, weight, range for the comfort, viability and tolerance zones].

The *type* is "basic", "essential" or "secondary" (see Section 4.1.3). It is inspired from the Maslow's hierarchy [Maslow 54]. The basic internal variables need to be satisfied first, and then come the essential ones and finally the secondary ones. This system gives a priority to the internal variable satisfaction. The *weight* is implied in the computation of the corresponding motivation value when the internal variable is in the viability zone. The *range* allows to define the limits of the different zones for the motivations subjective evaluation specific for each internal variable.

Motivation

[name, initial value, weights of the internal variables and the stimulus, hysteresis factor].

The *weights* are used to define the importance of the internal variables and the perceptions in the computation of the motivation value propagated in the hierarchy. The *factor* defines the hysteresis part, i.e., the motivation values which will be kept from the pervious iterations in the calculus of the current motivation.

Comportment

[location, corresponding motivation, name, source level, interaction factors, stimulus and motivation weight, intermediate and motivated action strength]

The *location* indicates the place where the object(s) needed to satisfy the corresponding motivation(s) is (are). The virtual human has to reach this location to perform the motivated action. If the *source level* is defined that corresponds to the possibility of having a dynamic quantity of the resource. For example, the food location has a certain quantity of food. When it is empty, the virtual human cannot satisfy his hunger and should go elsewhere. It is a list with an initial value, decreased factor and maximum value of the resource quantity. The *interaction factors* allow to choose preferentially by the action selection mechanism another behavior in order to chain them. For example, if the virtual human is in the exercise location and does push-ups, the corresponding effect will be to increase the resting behavior such as "go to the sofa to rest". Interactions should be defined with the following motivated action and the influence factor which determines the strength of the interaction. Indeed at 100%, the following behavior has many chances to be chosen and by default, there is no interaction (0%). If one interaction is defined in the comportment rules, it appears automatically in the interaction tab. The number of interactions is not limited. The *weights* are used to define the importance of the internal variable and the perceptions in the calculus of the comportments value propagated in the hierarchy. Finally the *strength of actions* corresponds to the criteria for designing an effective action selection mechanism by privileging the motivated actions over the intermediate actions [Tyrrell 93a]. In our model, the motivated action strength of the rule is twice the one of intermediate actions in order to be chosen preferentially by the action selection mechanism.

Action

[name, action type, corresponding internal variable, action length, action threshold, motivated action effect, distance and amplification factors of the perception, acceleration of the simulation, name of the keyframe, number and name of implicated objects, keyframe acceleration]

Four *types* of actions exist: actions with keyframes, actions without keyframes but with a location and the same for compromise actions. The *length* of the motivated action is in minutes and can more or less correspond to the real value which is much easier to define. The *action threshold* corresponds to the comfort zone but applied to the actions. Under this threshold, the actions aren't taken into account. This threshold can vary according to the situation and the emotions. The intermediate actions effect values, which increase internal variable values, are set automatically and randomly at the beginning of the simulation. The *motivated action effect* values, which decrease internal variable values, are defined in the rule. The *perceptions* of the virtual humans are separated into two factors:

- *the amplification factor*, which determines the evolution of perception values regarding the objects. In our case, the users can define the evolution factor of a logarithm curve; its maximum being when the virtual human is near the object.

- *the distance factor*, which specifies when the perceptions begin regarding the object (2 meters by default).

If the action can last long, such as sleeping, the simulation can be accelerated in order to avoid waiting too much until the virtual human wakes up. Finally, if the action has keyframe, different parameters have to be set such as the *name*, the *number* and the *name of objects* implicated in the keyframe and the *keyframe acceleration*. If the actions do not have keyframes, a position point can be set to specify where the virtual human should go in the environment to satisfy the corresponding motivation. The keyframe files should all be placed in a specific directory so that they are created and loaded automatically. Keyframes are not necessary for playing the simulation.

Compromise Action

[name, action type, compromise goal, corresponding internal variables, action length, action threshold, distance and amplification factors of the perception, acceleration of the simulation, name of the keyframe, number and implicated objects, keyframe acceleration]

The compromise actions have the same variables as the actions, except that their *location* has to be defined to know where they can be performed. By default, all the motivated actions implied in a compromise action can also be done separately. For instance, if the virtual human can drink and eat in the kitchen, he can also just drink or eat there. Finally, *corresponding internal variables* have to be defined to be decreased accordingly at the same time thanks to the compromise action.

These rules are used for propagating the activities of the internal variables and the perceptions through the hierarchical classifier systems according to the rule conditions. The choice is only made at the action level, following to the free flow hierarchy principle. The action selection mechanism finally chooses the most appropriate action according to the motivations and the environmental information.

New rules can be easily added if the syntax and the creation of the related rules are respected. The system will take automatically the new rules into account and they will directly be visible in the graphical interface. Most of the parameters cited above can be redefined and changed in real time during the simulation, thanks to a graphical interface

5.3 The graphical interface

During the implementation, we have realized that a fully automatic system obtained by learning is not very useful in our case and is difficult to implement because many parameters are very subjective to define. So, we prefer to implement a highly configurable model thanks to an initialization file or directly during the simulation. The graphical interface implemented in Tkinter [Grayson 00] directly in the python editor allows us to control the simulation in real-time. The evolution of the internal variables, motivations, and action values can be followed during the simulation, thanks to color code. The curves show not only the current values of the essential variables, motivations and actions, but also those from the previous iterations. This way, their evolution can be followed over time and one can control what the virtual human is doing. The graphical interface also shows the location, the name of the activated action and the current goal of the virtual human, in addition to the visualization in the 3D viewer. The point of view can be changed between a human top view and a general one. The pace of the simulation can be accelerated or stopped at any time by clicking on the space bar for a better observation. Two parameters of the perceptions can also be redefined in real time: the distance from which the virtual human detects the object(s) and the amplitude of the perception when the virtual human gets closer to the object or the location. Other parameters in the rules are specific for each object. As the system is highly configurable, nine tabs are available in the graphical interface (see Figure 5.5).

Motivation tab

The first tab corresponds to the configuration of the motivations (see Figure 5.5). The green gauge represents the overall level of the virtual human's motivations according to the threshold system with the comfort, tolerance and viability zones (vertical grey lines). Motivations are classified in three categories inspired from Maslow's hierarchy of needs [Maslow 54], i.e., basic, essential and secondary. The first parameter that one can change in real time and for each motivation is the effect of intermediate actions on the increase of the internal variables values. This parameter is defined randomly within a certain range at the beginning of the simulation. If one intermediate action effect is increased, the others are decreased accordingly to keep the same sum of the parameters for the coherence of the simulation. The second parameter is related with emotions but can be associated with motivations: It represents how much the virtual human likes/dislikes each motivation. For instance, he may like resting and sleeping and dislike doing sports and

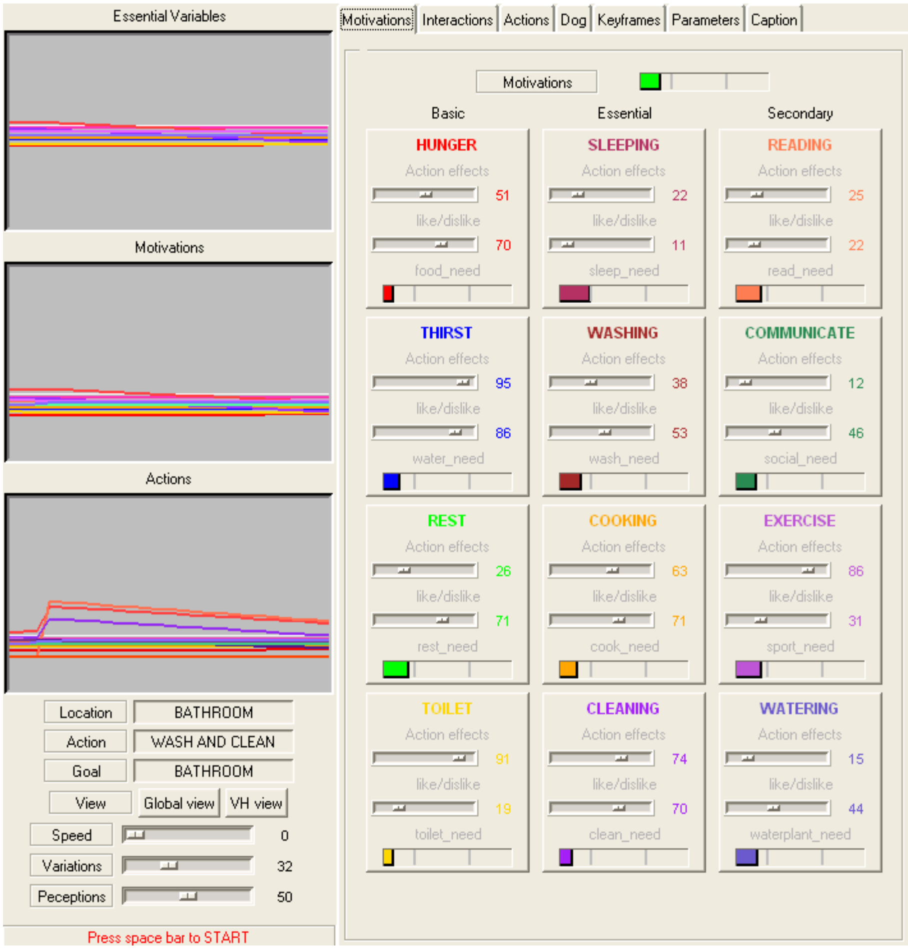

Figure 5.5: Overview of the application with the 3D viewer, the path-planning module and the graphical interface.

cleaning. In this case, he will privilege the actions correlated with the motivations that he likes. These parameters are defined randomly within a certain range at the beginning of the simulation like the intermediate action parameters. In the end, one can define a personality such as lazy, greedy, sporty, tidy, dirty, etc. for the virtual human. Finally each colour gauge shows the evolution in real time of the motivation according to the subjective evaluation of the motivations and the associated zones.

Action tab

Figure 5.6: View of the action tab.

The action tab allows to configure the action parameters such as their activation threshold, their length (expressed in minutes) and the perception factors for the location and the objects implied in the actions (see action rules in Section 5.2). The actions are separated in three types: original, additional and compromise. The original actions correspond to the independent ones in a specific location. The additional ones are the same actions as the original ones but at another location. For instance, if an original action is "reading in the sofa", an additional one can be "reading in the bed". Finally compromise actions imply several actions in order to satisfy several motivations at then same time implicating original and additional actions. It can be for example, "resting and reading in the bed". The configuration of the keyframes linked with the actions is possible in the initialization file or in the parameter tab.

Dynamic tab

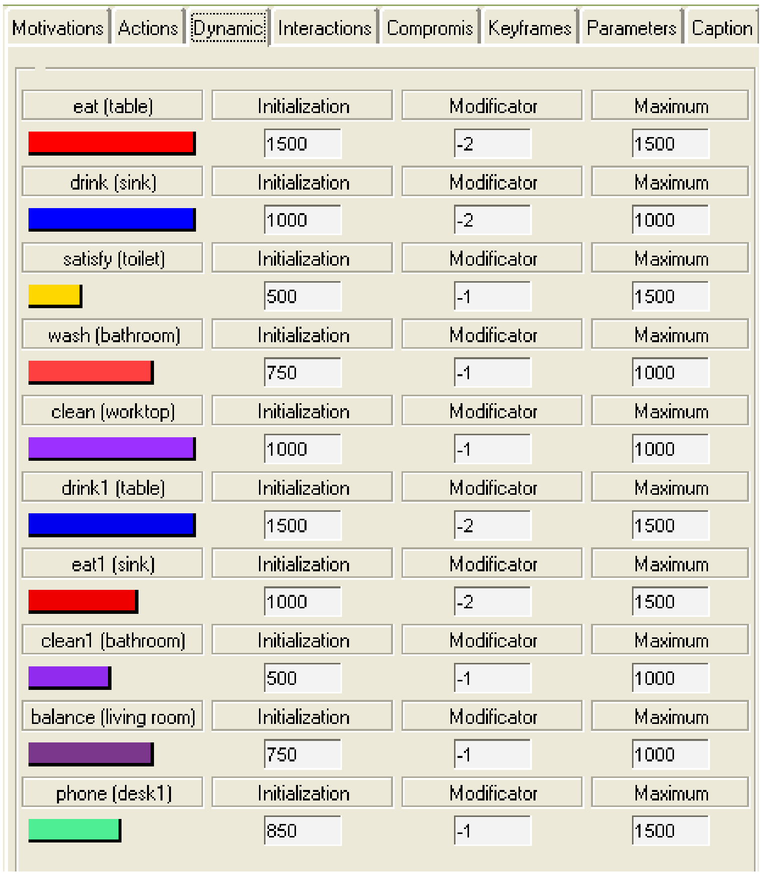

Figure 5.7: View of the dynamic tab.

Having a dynamic environment is important for a simulation [Cañamero 01]. Then, subjective or physical quantity can be defined. For instance, the quantity of food at each food location (represented in Figure 5.7 by a color gauge) can be set with the initial value, the decrease factor and the maximum value. If there is not enough food, the virtual human cannot satisfy its hunger and should go elsewhere. However for some motivations, it is more subjective such as cleaning or phoning. In this case, it can be not enough washing product or the telephone begins to be out of order. If this is defined in the initialization file, it will be automatically added in the dynamic tab.

Interaction tab

Figure 5.8: View of the interaction tab.

A dynamic environment implies interactions between behaviors because for instance if there is no food, cooking is necessary. If the phone is broken, the virtual human has to call the repairman with the second phone. The interactions correspond to a preference in the choice of the next behaviors by the action selection mechanism. This preference can be modulated by the interaction factor expressed in percentage. By default, there are no interactions (0%) and a very strong interaction should be defined with 100%. Chaining behaviors are then possible but the decision making keeps its flexibility. Complex and coherent behaviors can be generated easily by defining interactions and more unexpected and realistic situations can occur. When the interactions are defined in the comportment rules of the initialization file, they are automatically showed in the interaction tab.

Keyframe tab

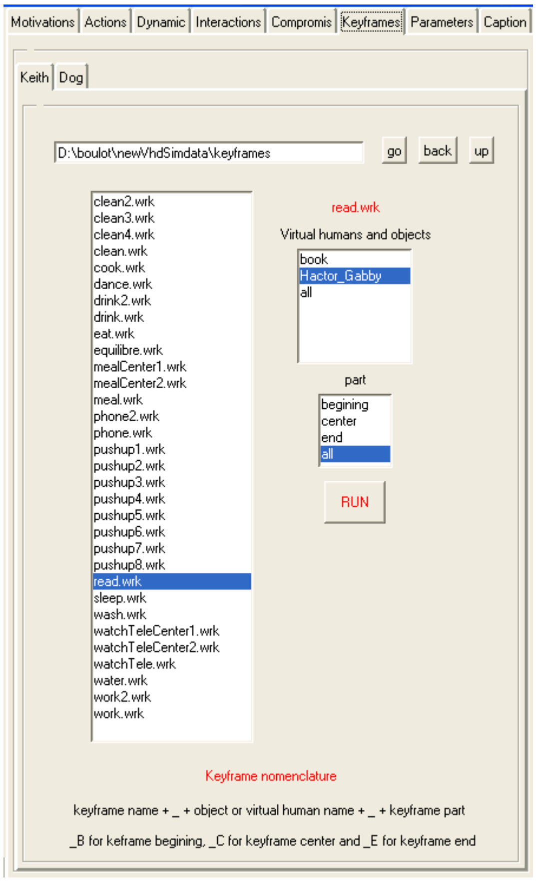

Figure 5.9: View of the keyframe tab.

In this tab, one can see and play separately keyframes associated with each character of the simulation. In our case, it corresponds to the virtual human named Keith and his dog. The keyframes for virtual humans are separated into three parts (see next Section). Thanks to this tab when the specific syntax for the keyframes is respected, they can be viewed in different ways: the entire keyframe or just the one part of the keyframe. Moreover keyframe parts of the virtual human and the object(s) can be showed separately. For instance, if the "read" keyframe is chosen, one can view the beginning part of the virtual human, the end parts of the book, all parts (beginning, center and end) of the virtual human or all together, etc...It allows to have a good control over the animation and to verify if the all keyframes work before starting the simulation.

Parameter tab

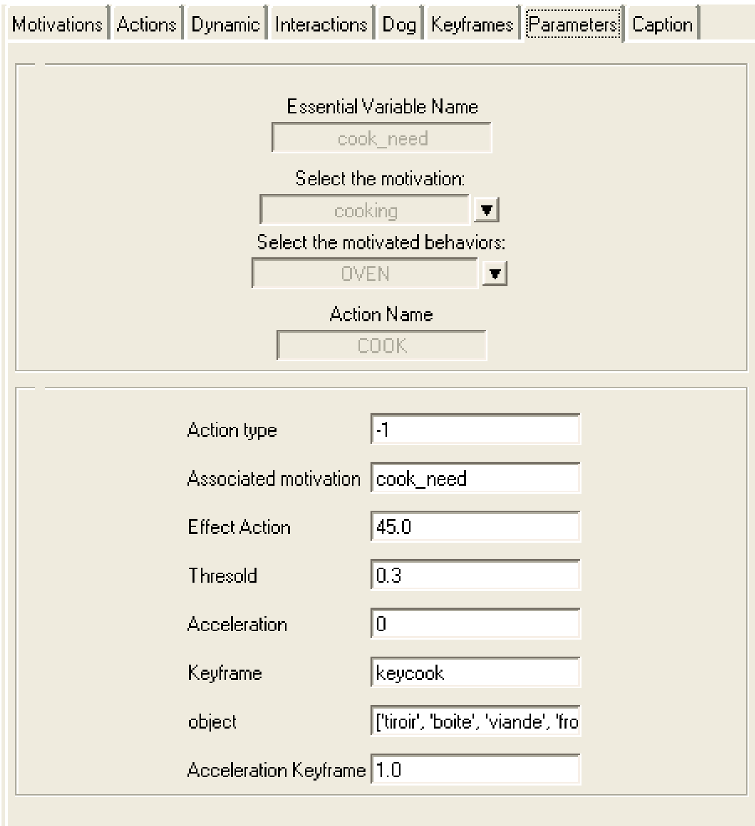

Figure 5.10: View of the parameter tab.

In the parameter tab, most of the parameters of the rules base can be redefined following the hierarchy links for coherent changes during the simulation. However, the parameters that can be reconfigured by users are accessible by other simpler tabs. This tab is for expert users only to have a fine tuning of the parameters and a precise control of the simulation.

Caption tab

With this tab, it is possible to view graphically all the levels of the action selection model with their interactions and dependencies. This tab is generated automatically based on the rule base of the initialization file. It helps to understand how the decision making works.

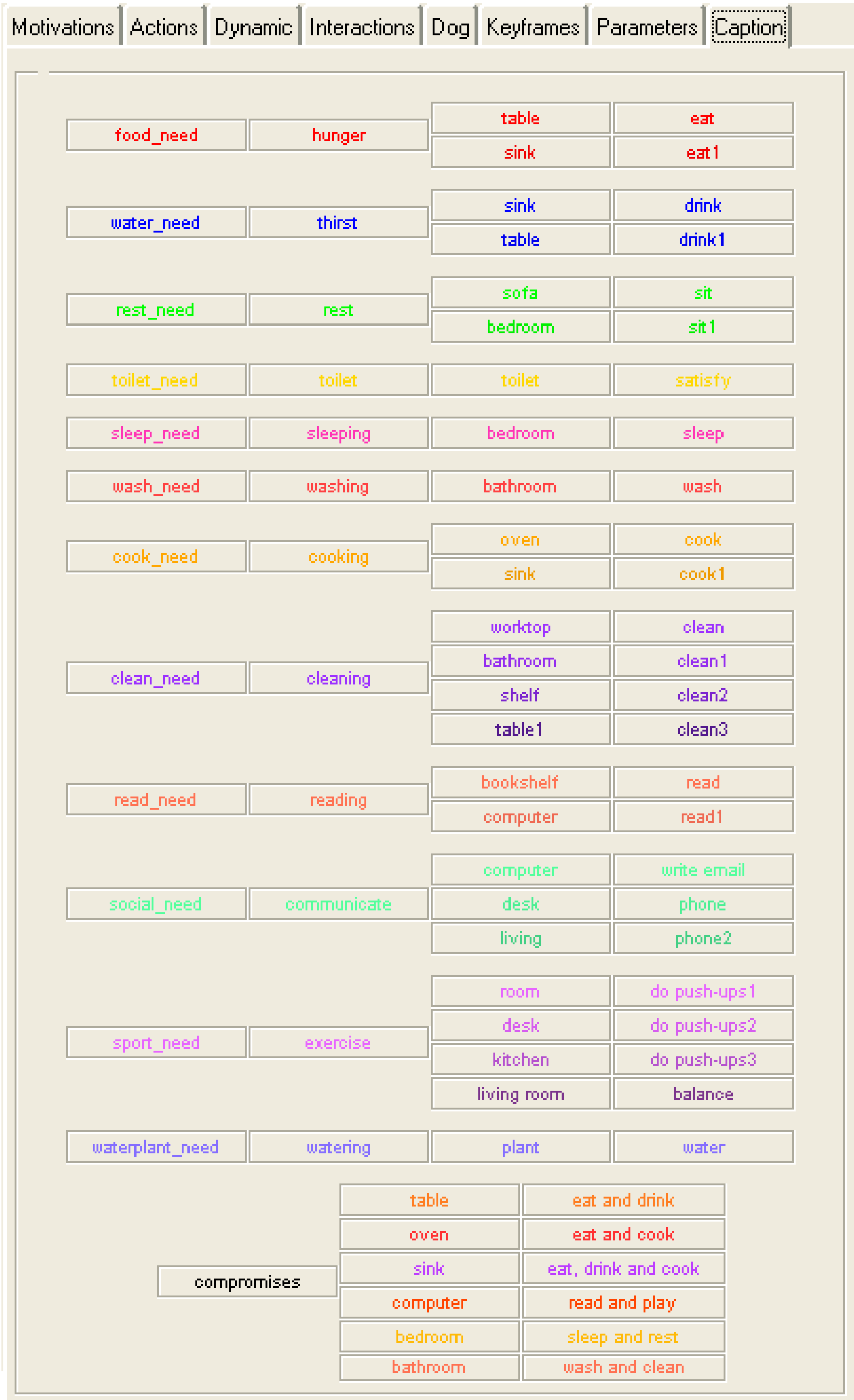

Figure 5.11: View of the caption tab.

5.4 Animation part

The keyframes and the environment are created in 3DSmax by the designers from the VRLab in a specific way to be easily integrated in the system thanks to VHD++. It allows to play keyframes on H-anim virtual humans and on objects, to display 3D scenes and virtual humans, to contol the animation and the simulation through the python editor, and to compute the path-planning for moving the virtual human in his environment by avoiding obstacles very useful for implementing a action selection mechanism for autonomous virtual humans in persitent worlds.

The path-planing module

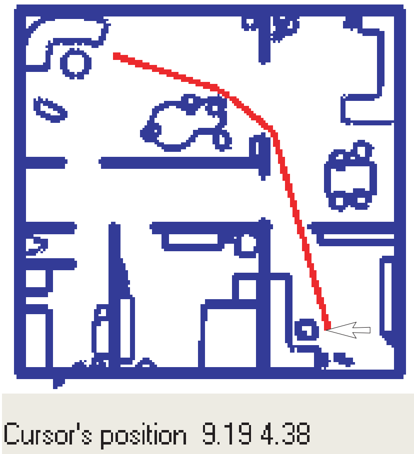

Figure 5.12: View of a path generated by the path-planning module

The path-planning module was developed by Kallmann [Kallmann 03b] and allows the virtual human to navigate in his environment. An XML file is automatically generated from 3DSmax with all the obstacles coordinates and loaded in the path-planning module (blue in Figure 5.12). Once the initialization is done, it generates a path in real-time (red in Figure 5.12) in the form of a list of points between the location where the virtual human is and the one where he should go. However, for following this points sequence using the walk engine [Boulic 04], we adapt and enhance these techniques so that the virtual human can reach specific locations to satisfy motivations, avoiding obstacles on the way.

Keyframes

Keyframes are chosen to simplify the animations of actions. However, they can be replaced by smart objects developed in the VRLab [Kallmann 01, Kallmann 03a]. Some actions don't have keframes and images representing the corresponding action are showed instead. But when keyframes are designed, they must have three different parts named:

- The pre action part allows to prepare the virtual human to perform the action keyframes. For instance, if he wants to drink, he has to take the glass of water and brings it to his mouth.
- The action part corresponds to the motivated action. Only at this stage do the corresponding internal variables decrease. For instance, when the virtual human has caught the sponge, he can begin to clean the table until he estimates that he has cleaned enough.
- The post action part represents the moment when the virtual human has finished performing the motivated action and he has to put back the object it was using. For instance, when the virtual human has finished it, he has to hang up the telephone and place it back where it was before its action.

When keyframes are created, a specific syntax has to be respected:

Name_Actor_Part.wrk

with *Name* is the name of the keyframe, *Actor* represents the object or virtual human name and *Part* corresponds to the keyframe part which can be *B* for pre action part, *C* for action part or *E* for the post action part.

At the end, these keyframes should be put in the keyframe directory in order to be created, loaded and played by VHD++ (see Figure 5.9). It allows to simplify the management of the keyframes by the system. During the simulation, the system should verify the synchronization of the keyframes, i.e., if different parts are finished and particularly post action keyframes before taking into account another motivation. Thanks to the keyframe tab, the keyframes can be viewed separately or globally for the virtual and the objects. Moreover, the goal positions are extracted automatically from the keyframe file and added to the corresponding comportment rules. Then, when new keyframes are added, the virtual human can directly perform the action.

However some problems persist between path-planning and keyframe positions. The path-planning algorithm has a safety distance (see Figure 5.13) when it computes the

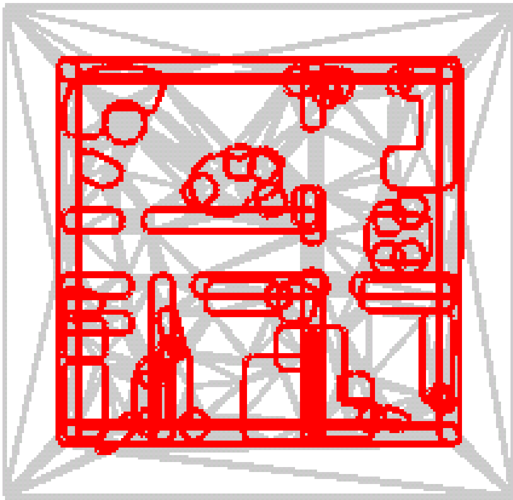

Figure 5.13: View of path-planning module with the security distance.

paths in order to avoid the virtual humans to be hurt by walls. This safety distance prevents the virtual human to get to the keyframe positions to perform motivated actions. So, a specific algorithm has been developed so that virtual humans can reach the position where keyframes can be played with possible associated objects inside this path planning security distance. The algorithm manages also when the end of the keyframes because the virtual human is still in the security zone. It allows the virtual human to come out of this zone in order that the path-planner can generate a plan to reach a new location.

The action positions which are not linked with a keyframe can be modified in the initialization file or in the parameter tab and an action position should be defined instead. Indeed, when a keyframe should be played, if it involves objects, then the position cannot be changed. However, action positions which are not associated with keyframes can be moved wherever the user wants according to the path-planning zones. The coordinates of a position can be found easily, thanks to the path-planning visualization that indicates the coordinates where the mouse is pointing (see Figure 5.13).

Animation control

Loading and playing keyframes, modifying the 3D camera and the position of virtual humans and objects, etc. can be done at every moment with simple commands thanks to the python editor of VHD++ or directly during the simulation with the graphic interface.

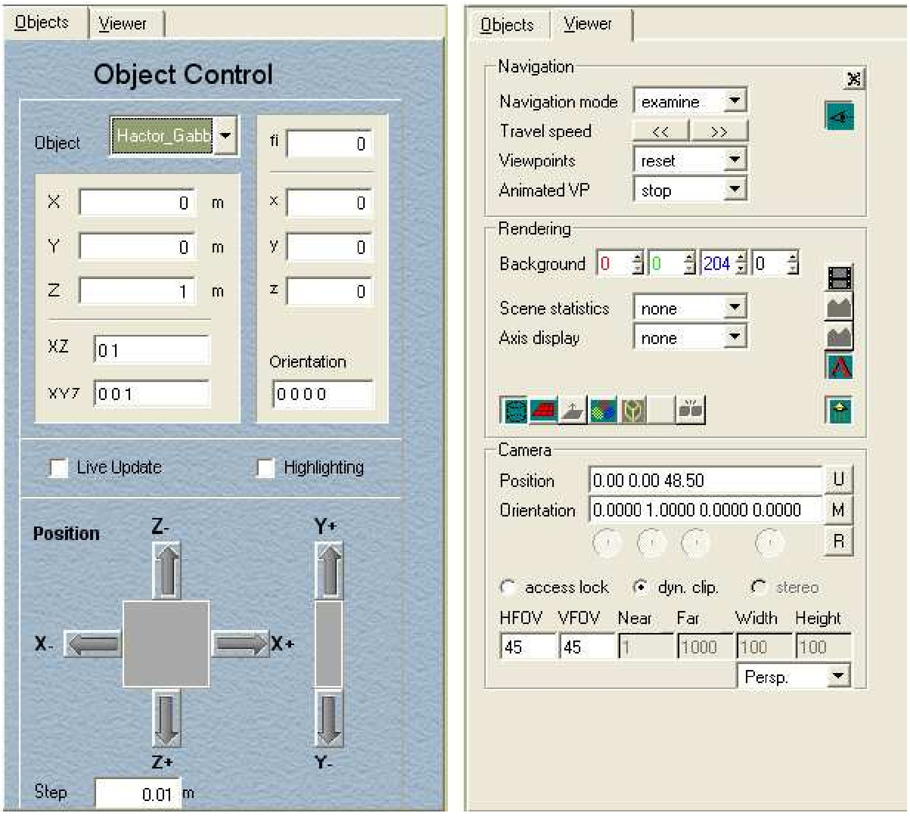

Figure 5.14: View of viewer and object control tab.

5.5 Summary

We use VHD++ [Ponder 03] for implementing and testing our model of action selection for virtual human in persistent worlds but it can be used on other platforms such as video game engines or robots. VHD++, which is developed in the VRLab, allows us to focus on our model by providing all the virtual human functionalities that we need such as walking, playing a keyframe, etc. Many parameters in the rules can be defined in the initialization file and thanks to the graphical interface during the simulation in order to modulate and influence the behaviors of the virtual humans at every level of the action selection model. With a highly configurable graphical interface, the users, and in particular psychologists, can test all their ideas by tuning the architecture as they want. It allows also giving some personalities to virtual humans such as lazy, greedy, sporty, tidy, dirty, etc. A fully automatic architecture by learning is not possible in our case because many parameters need a subjective tuning depending on the user, the environment, and the goal of the simulation.

Chapter 6

Test Scenario

The decisions of the virtual humans should be made in real time and not cost too much, because the graphical rendering of the simulation takes a lot of computer processor. The decision model should be quick and as simple as possible to run in real time. However the behaviors should be realistic and coherent.

Autonomy requires individuality. If the virtual humans have to make their decisions by themselves in real-time, they need to be strongly autonomous [Luck 01] and distinctive. This corresponds to our "bottom-up" approach, i.e., first understand and design a distinctive autonomous virtual human, which needs already much studies instead of having several simple agents with social interactions. The aim of this work is to study individuality and autonomy in order to obtain more interesting and realistic autonomous virtual humans in persistent worlds. To test our hierarchichal and behavior-based model of action selection, a complex environment with many motivations had to be designed. The number of tasks, possible conflicts, and interactions between them are fundamental factors to the complexity of a scenario. The relation between the durations of tasks on the one hand and primitive operations on the other hand is an indicator of the needs for high-level behaviours such as planning and the use of hierarchy in the organization of behaviours [Aylett 96].

6.1 The actor

We choose to simulate one young virtual human, whose name is Keith. Our action selection architecture gives him a strong autonomy and individuality, i.e., the possibility of living by his own in his environment in real-time without time limit and external interventions according to his motivations and perceptions.

Figure 6.1: View of keith, the virtual human used in the simulation

6.2 The motivations

We arbitrarily define twelve conflicting motivations that a human could have in this environment with their specific goals and the associated motivated actions. They are described in the Table 6.1, in order to represent a sample of motivations that a human could have in the same conditions. None of the motivations are linked with social interactions except maybe phoning according to our approach. The motivations are defined accordingly to the application context. For instance, if it is a working office, the motivations will be associated with this environment: writing a report, phoning to a client, etc. The model is generic and the users have to adapt it to their simulation (virtual or robotic).

Each motivation comes from one and only one internal variable. They are divided into three categories: basic, essential and secondary, inspired from Maslow's hierarchy of needs [Maslow 54]. This distribution allows to have several types and priorities for the motivations. First, all basic motivations are essential to all living beings and then have the highest priority of satisfaction: hunger, thirst, rest and toilet needs. Then

motivations	goals		actions	
hunger	table	1	eat	1
	sink	2	eat1	2
thirst	sink		drink	3
	table		drink1	4
resting	sofa	3	sit	5
	bedroom		rest	6
toilet	toilet	4	satisfy	7
sleeping	bedroom	5	sleep	8
washing	bathroom	6	wash	9
cooking	oven	7	cook	10
	sink		cook1	11
cleaning	worktop	8	clean	12
	shelf	9	clean1	13
	bathroom		clean2	14
	table1	10	clean3	15
reading	bookshelf	11	read	16
	computer		read1	17
communicating	computer	12	communicate	18
	desk1	13	phone	19
	living	14	phone2	20
exercise	room	15	do push-up1	21
	desk	16	do push-up2	22
	kitchen	17	do push-up3	23
	Living room	18	balance	24
watering	plant	19	water	25
compromises	table		eat and drink	26
	sink		eat, drink and cook	27
	computer		read and communicate	28
	bedroom		sleep and rest	29
	bathroom		wash and clean	30
default	sofa		watch TV	
...	...		...	

Table 6.1: The twelve motivations with associated locations and actions

come the essential motivations such as sleeping or washing and finally secondary or the fun motivations which have the lowest priority: reading, playing with the computer or watering the plants.

Each motivation has one or several specific locations where the virtual human can satisfy it. In the environment, there are sixteen locations, enumerated in the Table 6.1. That is why some motivations have more than one location. The virtual human can also satisfy several motivations at the same place; altogether or separately. For each different place, where one motivation can be satisfied, the corresponding action is not the same and should be configured independently from the others. Moreover, compromise actions increase the number of actions. For clarity purposes, we group compromise behaviors according to the locations. However, thanks to the name of the actions, the corresponding motivations satisfied by the compromise behavior can be found. The caption tab shows these links to have a better idea (see the figure 5.11).

We have chosen a pleasant default action which is watching television. It corresponds also with the reality, most people do this during many hours during the day. When they have nothing to do, they watch the television. The default action can also be working on the computer, surfing on the web, reading books, etc. and is activated when all the motivations are satisfied, i.e., are in their comfort zone. Indeed each action has also a specific activation threshold related to the subjective evaluation of the motivations. Their values should overcome this threshold in order to be activated by the decision-making system. The thresholds can be modulated by the emotions, i.e., if the virtual humans like or dislike the actions.

In the end, the action selection model has many conflicting possibilities for driving the behavior of the virtual human and will choose the most appropriate one according to internal and external factors by planning coherent and flexible behaviors. Moreover, the number of motivations, goals, and actions is not limited. The model is not dependant of this environment or this configuration. It can work in other environments and with other agents such as a dog (see Section 7.5).

6.3 The environment

The environment should have several characteristics to be able to test the action selection model: dynamism, uncertainty, threats, availability of resources, etc. [Cañamero 01]. However the aim of this work is to show that, in predefined environments such as video games for instance, virtual humans can live their own lives according to their motivations

Figure 6.2: View of the apartment in the 3D viewer of VHD++

and emotions. They have also to adapt to their environment, which can cause unexpected or opportunistic situations. In our case, the survival functions are not essential and the management of threats is not our priority. However, the model can manage them.

We choose a 3D apartment, in which the virtual human can "live" autonomously by perceiving his environment and satisfying different motivations at specific goals (see table 6.1). In an apartment, the social interactions are not necessary as in a work office or sport club. We can let him in this environment and he will choose what he wants to do continuously. People do not stay all the time in a work office, sport club or discotheque. In an apartment, the virtual human can work, play, and satisfy his basic, essential and secondary needs and so on in the same place. Moreover these three categories of motivations can be easily defined. An apartment is not so complicated to design in 3D. This environment has all the necessary conditions to test our action selection model for autonomous virtual humans in persistent worlds.

6.4 Summary

The choices of actions in this environment are not trivial because they depend on many parameters and several options are available: the environmental changes, the opportunistic behaviors, the conflicting motivations and locations, the compromise behaviors, the resource quantities, the emotion influences, the behavior interactions, the "subjective" evaluation of motivations, the hysteresis, the distance of goal, the rule weights, the unexpected situations such as danger, the different action plans and their possible interruptions. Another strong constraint is that the decisions must be taken in real time. This implies that the behavior planner has to be quick. Moreover, to be efficient in this environment, the decision-making has to be coherent, i.e., managing persistence of actions, time-sharing, and behavior plans and also flexible, i.e., offering the possibility of interrupting plans, compromise and opportunist behaviors. In the end, this scenario is complex enough to test all the functionalities of the action selection model of distinctive autonomous virtual humans in persistent worlds.

Chapter 7

Results

The results show that our model of action selection is flexible and robust enough for designing in real-time autonomous virtual humans in persistent worlds so that they can live their own life according to their internal and external factors. The model also integrates some subjective evaluations of the decision-making to obtain more realistic and believable behaviors. Finally, some tests are made for demonstrating the generality of the model.

The test simulations have been realized over 65000 iterations or one hour and a half. We estimate that it is long enough to prove the efficiency of the decision system in persistent worlds. We also have let the simulation working during 9 hours but the results are difficult to exploit because of the data size (433 385 iterations). Moreover, the differences between a 9h simulation and a 1h30 one are not important enough to be worth the time spent. The following screenshots of the graphical interface show the evolution of the internal variables, motivations and actions during the simulation. It is only for clarity purposes that some results are based only on two motivations and that the shown cases are often extreme to better illustrate the functionality results. In normal conditions, the action selection model manages thirty actions (see Table 6.1).

7.1 Reference simulation

To have a base to compare the performances of the model functionalities, the action selection system has been tested with only the basic functionalities (see Section 4.2), i.e. identical action durations and motivation evolution parameters, as well as no dynamic quantities, motivation interactions, compromise behaviors, second actions in a different or in the same locations and emotions. The decisions depend principally on the evaluations of the motivations and the perceptions made by the action selection mechanism. Indeed, each internal variable has only an associated motivation, location and motivated action (see Table 7.1).

(1)	food_need	hunger	table	eat
(2)	water_need	thirst	sink	drink
(3)	rest_need	rest	sofa	sit
(4)	toilet_need	toilet	toilet	satisfy
(5)	sleep_need	sleeping	bedroom	sleep
(6)	wash_need	washing	bathroom	wash
(7)	cook_need	cooking	oven	cook
(8)	clean_need	cleaning	worktop	clean
(9)	read_need	reading	bookshelf	read
(10)	social_need	communicate	computer	write email
(11)	sport_need	exercise	room	do push-ups1
(12)	waterplant_need	watering	plant	water

Table 7.1: Visualisation of the hierarchy of the reference simulation

As all the parameters are equal, the virtual human executes each action approximately the same amount of time (see Figure 7.1), (i.e.,13 times), during 65000 iterations or 1.5 hours of simulation. The differences come principally from opportunistic behaviors. The virtual human has achieved a total of 156 actions during the simulation. As he takes his decisions at each iteration, it corresponds to a huge number of choices. The action selection model does not mindlessly pursue an action indefinitely to the detriment of other needs. This implied that the model shares well its time between all the conflicting motivations (time-sharing) in spite of the their priorities (see Section 4.1.3).

The presence of the virtual human in the different locations (see Figure 7.2) confirms that the virtual human shares well his time between the possible actions because the time passes at each location are almost equal in spite of the difference of distance between the goals. The percentage of time the virtual human is in one location corresponds almost in this case to the percentage of time the virtual human performs the associated motivated action in this locations. Otherwise the virtual human spend 5% of his time watching television (default action). The rest of the time (35%) has been used to move in the apartment to reach specific locations.

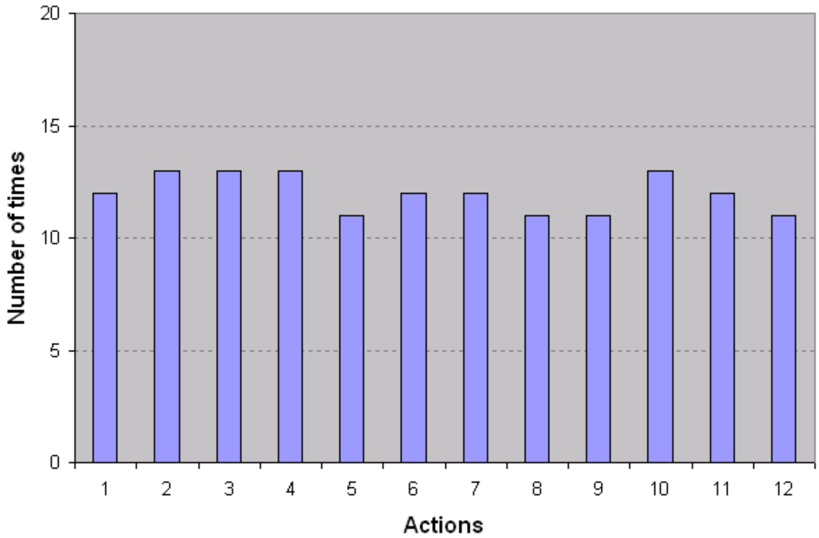

Figure 7.1: Number of times that each action is chosen by the action selection mechanism over 65000 iterations

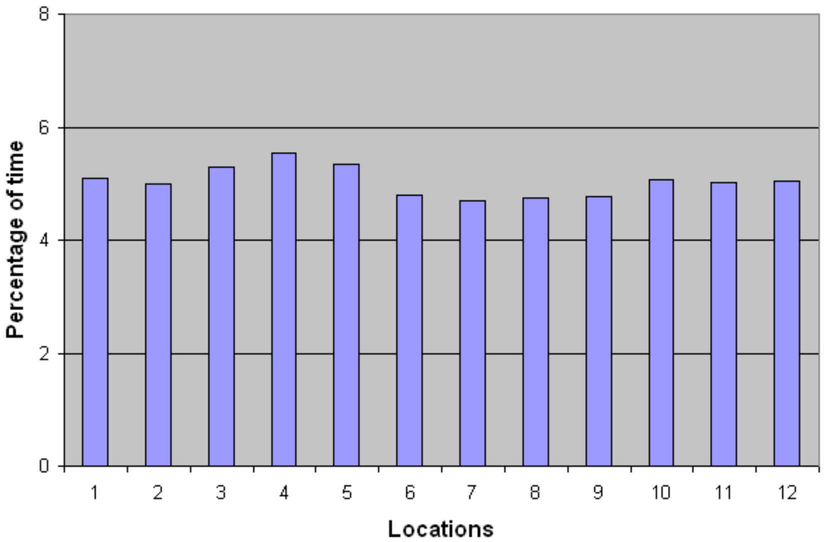

Figure 7.2: Percentage of time the virtual human is in different locations over 65000 iterations. He was at the defauflt location (TELE) during 5% of his time and in movement during 35% of his time (see Table 7.1 for the corresponding locations to the numbers)

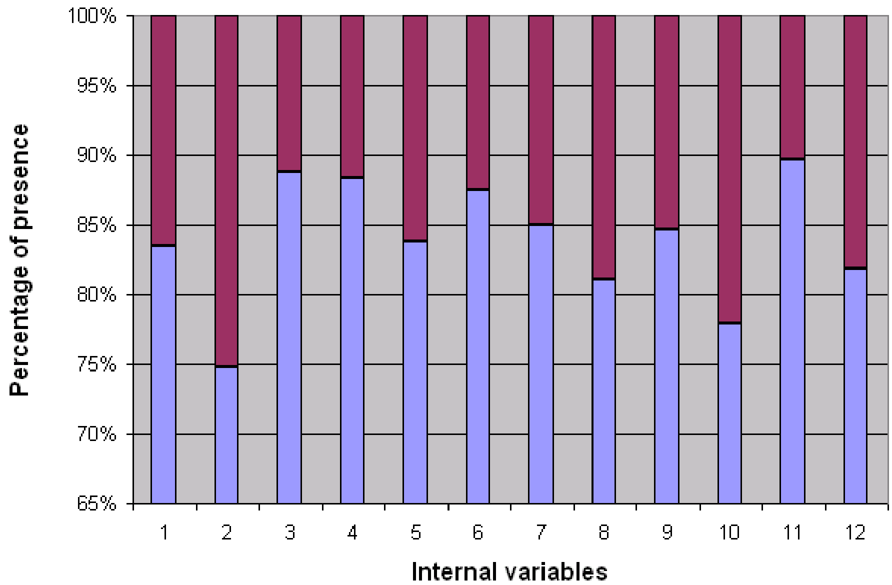

Figure 7.3: Percentage of internal variable presence according to the threshold system: comfort (blue), tolerance (brown) and viability (white) zones

With this model configuration, the robustness of the decision-making system is directly tested because all motivations are conflicting, i.e. their evolution and length are the same. The decision-making system has to satisfy them altogether. However, the action selection mechanism manages its task even when many motivations are high at the same time as for instance, at the beginning of the simulation (all the motivations arrive to the tolerance zone at the same time). In the end, the action selection mechanism has an efficient persistence of actions assumes its main role because it maintains the internal variables in their comfort zone (see Figure 7.15) during about 80 % of the time. It is enough to show interesting results concerning continuous decision-making for virtual humans in persistent worlds. In the next section, we describe the advantages of the model functionalities in comparison with this reference simulation.

7.2 Flexible and reactive architecture

To be reactive, an action selection model needs to take into account in real-time the information coming from the environment and adapt its decisions accordingly. This allows to have less predictable behaviors. The virtual environment has to be complex enough to authorize opportunist and compromise behaviors as well as unexpected situations and behavioral interruptions. It also has to be dynamic with variable quantity of resources and varied with many locations everywhere in the apartment for satisfying motivations. To perceive all these functionalities, the virtual human needs a perception system. In our model, the perception system is simple and detects objects or obstacles in a certain distance defined by the users (by default 2 meters). The perception value evolves exponentially from the distance limit until the location where the virtual human should be. The perception distance and evolution can be specific for each object if necessary. A more complex sensory system could be integrated easily but it is beyond the aim of this thesis.

7.2.1 Location distance

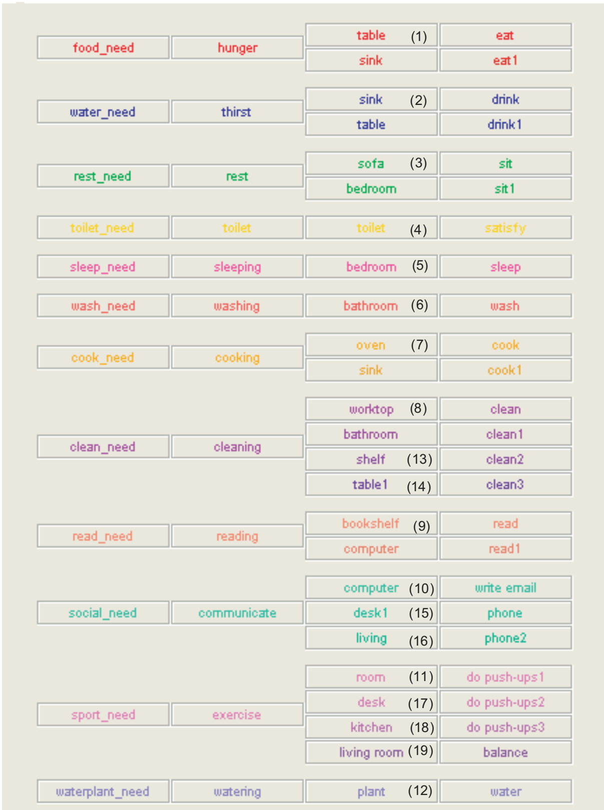

Figure 7.4: Visualization of hierarchy for the location distance test

As some motivations can be satisfied at several locations, there is a conflict between them. In this case, the behavior values should be modulated according to compromise behaviors, the distance to the goal location and the available quantity resource. The distance to the goal location should also be taken into account because the virtual human should go to the nearest location except in the case of a compromise behavior, or if the quantity of the resource is too low. Indeed, if a compromise behavior is possible, it should be chosen preferentially [Tyrrell 93a] because it satisfies several motivations at the same time and costs less in the decision-making thus. If the quantity of the resource is low, the virtual human cannot satisfy his motivations enough. In this case, he will choose a more distant location where the internal variables can return to their comfort zones.

To test the influence of the perceptions on decision-making, seven new locations where the virtual human can satisfy the existing motivations have been added (13 to 19, see Figure 7.4 for the correspondence between the numbers and the locations).

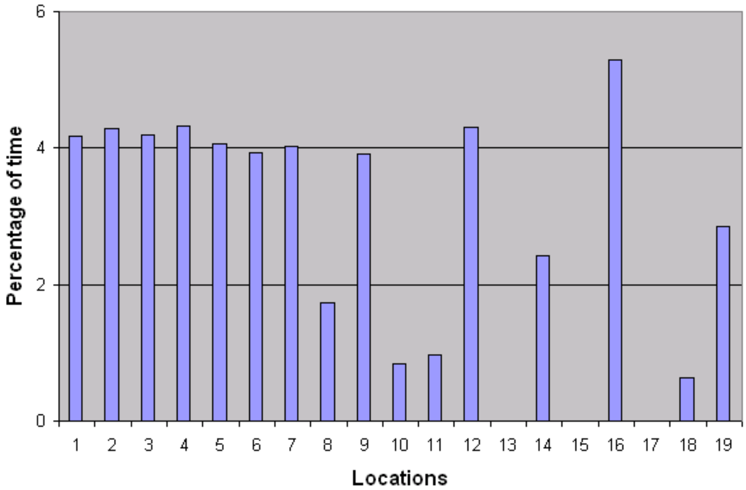

Figure 7.5: Percentage of virtual human presence in the different locations

In Figure 7.5, the locations 14, 16 and 19 are closer to where the virtual human is most of the time than the ones in the reference simulation (8, 10, and 11). The virtual human goes preferentially to the closest location because the distance is taken into account in his choice and the perception system increases the value of the actions when he passes near it. In Figure 7.5, the virtual human is more often in the locations 14, 16 and 19 because they are in the same room as the one where the default action is achieved. The locations in the desk room (10, 13, 15, and 17) are neglected. For the other locations, e.g. the toilets or the plants, it changes nothing because there are no conflicting possible actions to satisfy the motivations and the virtual human has to go there anyway. In the end, the action selection model takes into account the distance of the locations when conflicting actions to satisfy the same motivations are possible at different locations.

7.2.2 Behavior interruptions

The decision-making of the virtual human must be coherent and robust. However, it should also manage behavior interruptions (see Section 4.2.7) to have a greater flexibility in the action selection mechanism. When it happens, useless plans that don't match with the environment changes should be abandoned and new ones should be generated. It allows opportunist behaviors and situatedness to the virtual human. After the interruptions, the decision-making should resume the normal task.

Figure 7.6 shows a behavioral interruption due to the different evolution of the motivation parameters during the test simulation. Hunger is the highest motivation at the beginning and a behavioral sequence of intermediate actions has been generated to reach the food location and activate the "eat" action. However, the hydration state exceeds the nutritional state at t_0 due to their different parameter evolution, and thus thirst overtakes hunger. In this case, the action selection model stops the current "reach the food location" behavior and generates a new behavioral sequence of intermediate actions to reach the drinking location and satisfy thirst. After the water need is back in its comfort zone, the decision-making system will resume its previous actions in order to satisfy hunger. Opportunist behaviors can also interrupt the current behavior, as shown in Figure 7.7. The action selection model is flexible enough to adapt to the internal and external factors and thus the most appropriate action is always chosen.

7.2.3 Opportunist behaviors

Opportunist behaviors (see Section 4.2.2) are necessary in every environment to give dynamism to the behaviors. They are used to test if the action selection mechanism is reactive and situated following in real-time the changes of the environment. In our model, action positions can be changed if necessary in the rules or in the parameter tab (see Figure 5.10). In this case, the action selection mechanism will take it into account and adapt its choice accordingly, as explained in the example above.

Figure 7.7 illustrates an opportunist behavior. The virtual human is thirsty and goes to a drink location. He doesn't take his hunger into account because the corresponding nutritional state is within the comfort zone. However, when the virtual human passes near food, an opportunist behavior occurs, generated by his perception system. Hunger and the associated behavior are therefore increased proportionally to the distance to the food source. So, the "eat" action value happens to exceed its activation threshold at (t_0). Finally the "eat" action is chosen by the action selection model even though the

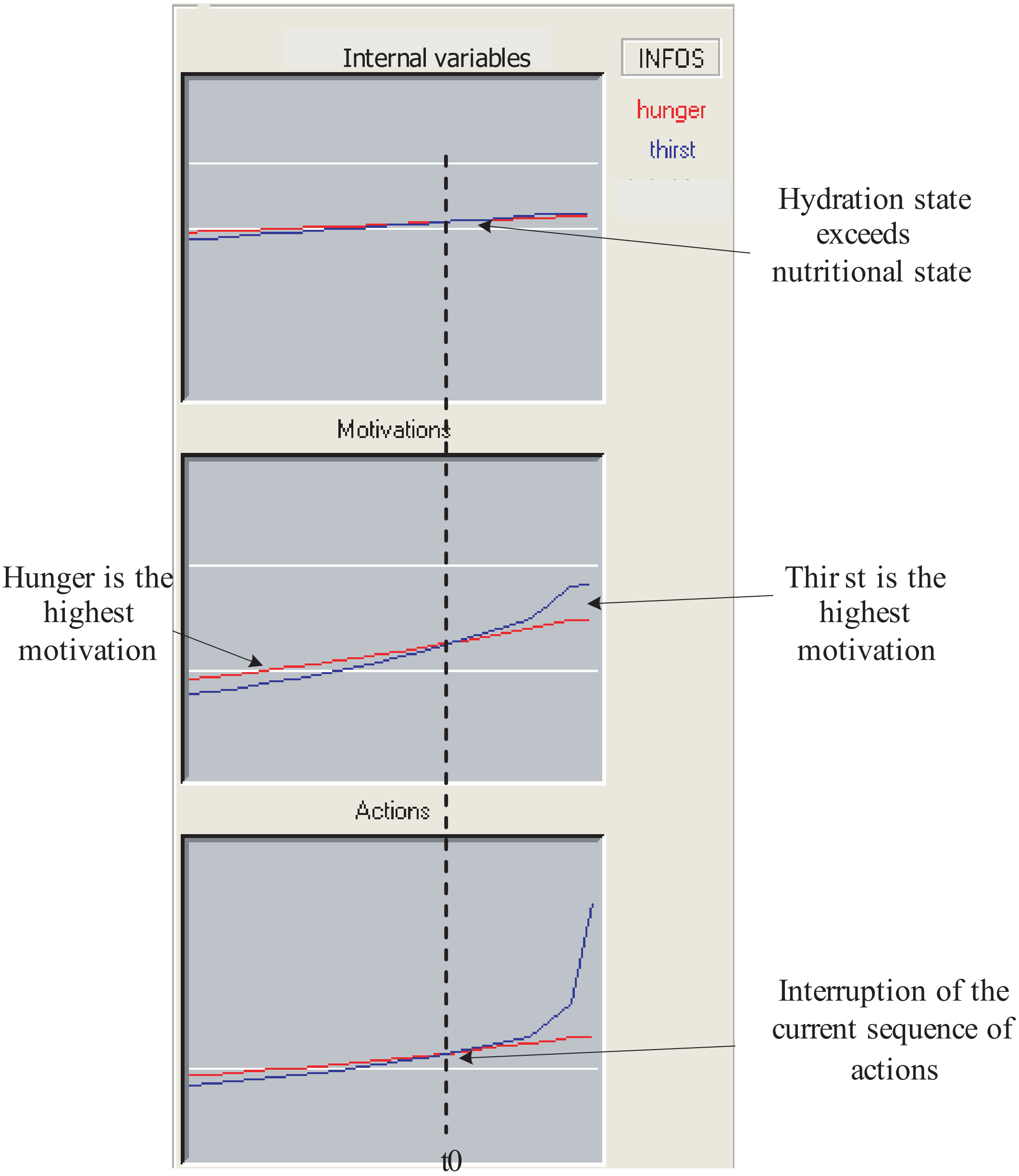

Figure 7.6: Interruption of the current behavior by another motivation more urgent to satisfy

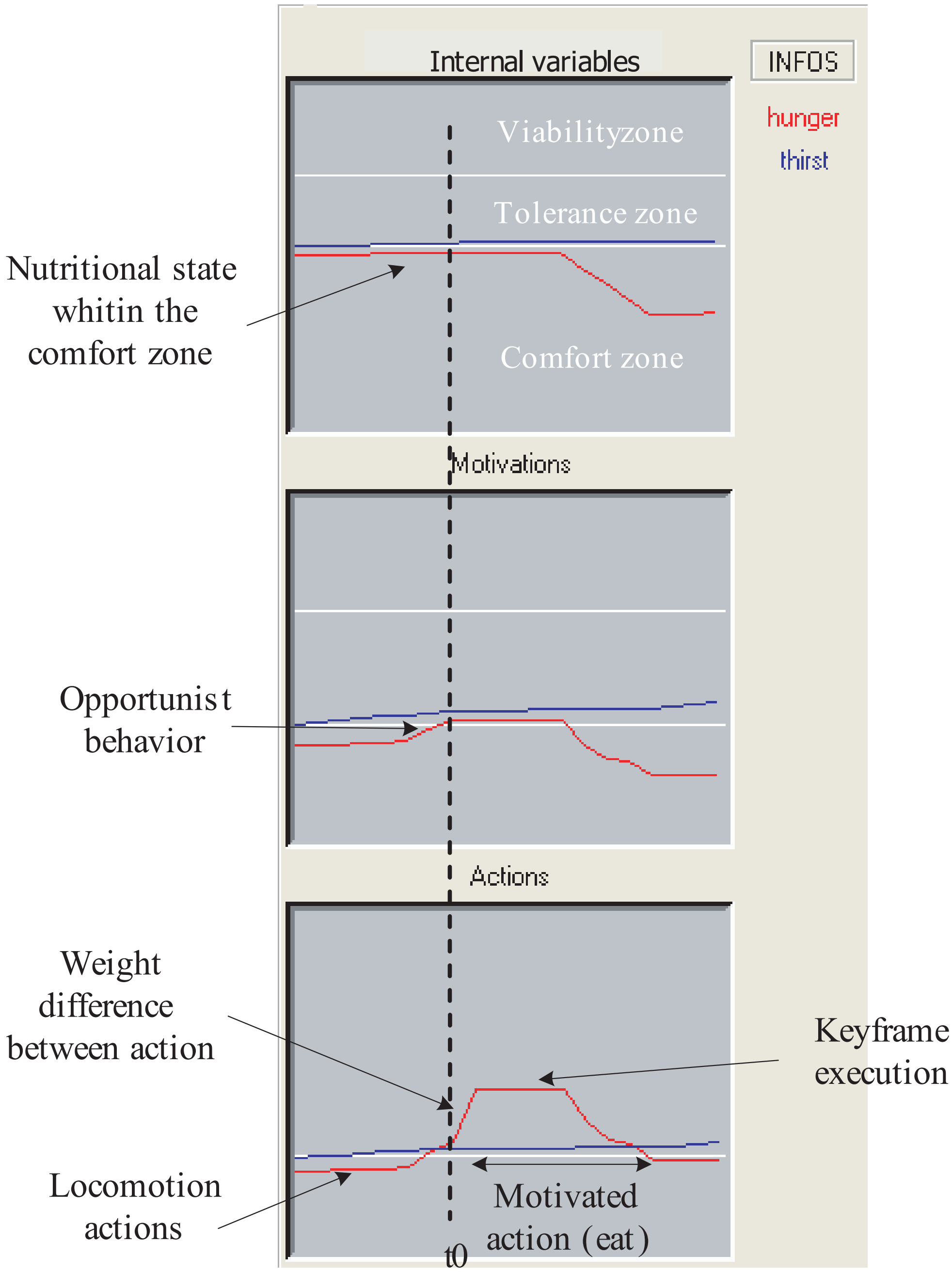

Figure 7.7: Opportunist behavior when the virtual human passes near food on the way to the drink location

nutritional state is within its comfort zone and the hunger is not the highest motivation. With opportunist behaviors, the virtual human is situated in his environment and highly reactive to its changes.

7.2.4 Compromise behaviors

Compromise behaviors (see Section 4.2.6) are very advantageous for the decision-making system because they allow to satisfy several motivations at the same time instead of going to several locations. Indeed, the role of the action selection mechanism is to maintain the conflicting motivations low. Compromise behaviors are possible in the apartment (see Table 6.1) but has to be defined in the initialization file. All actions performed at the same place can be grouped in a compromise behavior. The different actions can be done also separately. Indeed to activate the compromise behavior, all values of the actions involved must have overcome their activation thresholds. Compromise behaviors are in priority chosen by the action selection mechanism according to the ethological criteria [Tyrrell 93a]. For instance, at the desktop, the virtual human can read some papers or communicate by sending e-mails, but he can also do both of them in a compromise behavior satisfying all the associated motivations at the same time. In the apartment, five compromise actions have been defined at different places (see Table 7.2).

	table	eat and drink	(26)
	sink	eat, drink and cook	(27)
compromises	computer	read and play	(28)
	bedroom	sleep and rest	(29)
	bathroom	wash and clean	(30)

Table 7.2: Visualization of the hierarchy for the compromise behaviors

Figure 7.8 illustrates a compromise behavior. The highest motivation is hunger and the virtual human should go to a food place where he can satisfy this need. However, at another location, he has the possibility to both drink (green line) and eat (red line). Since thirst is quite high and the action selection model prefers to choose compromise behaviors, the compromise action "eat and drink" (black line) is chosen and the virtual human goes to this location even though it is farther. Indeed he can satisfy both food and water needs at the same time, instead of going first to a closer food location and then to the closest water source (blue line). As long as either the food and the water

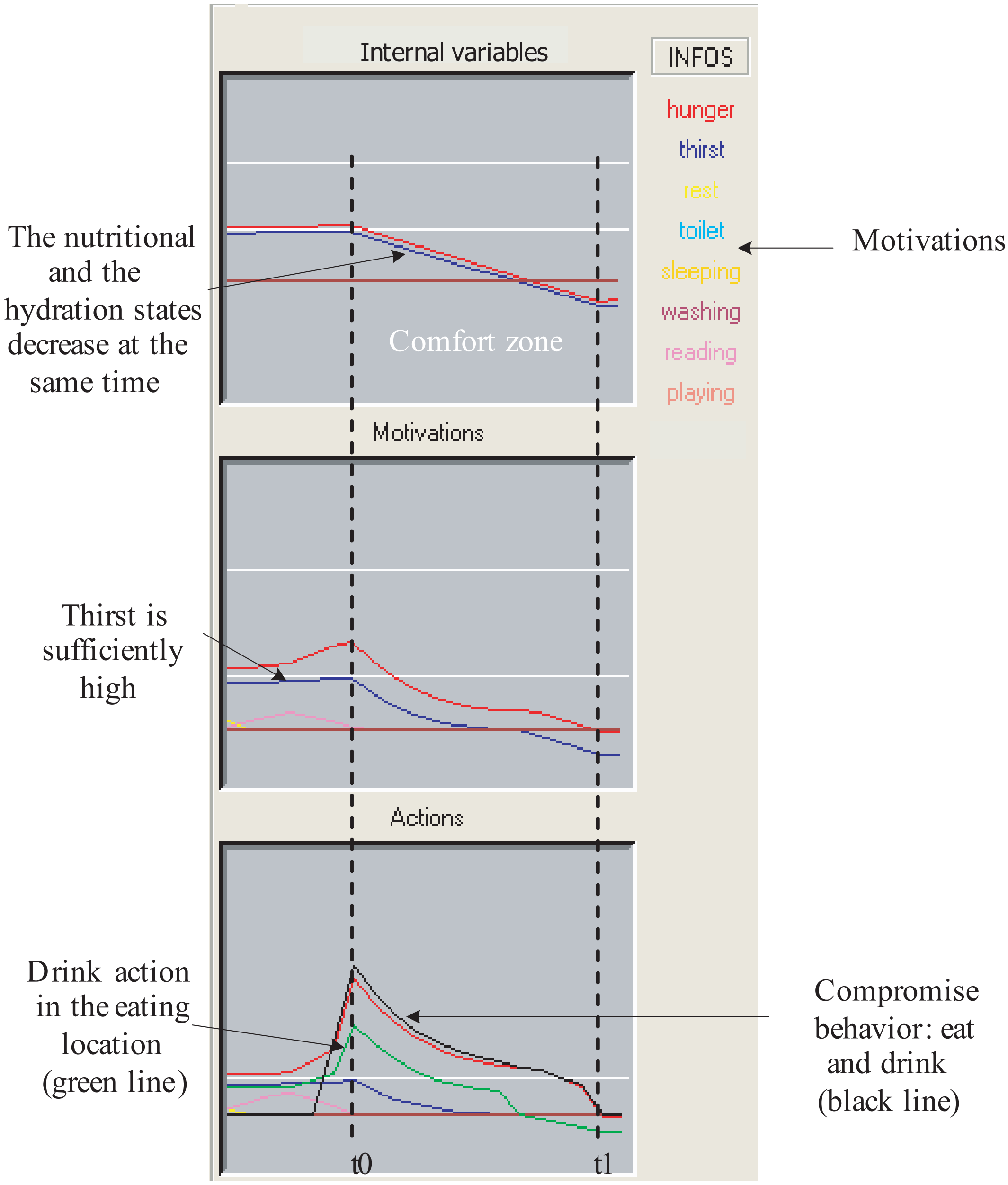

Figure 7.8: Compromise behavior "eat and drink" which decreases the food and the water needs at the same time.

needs have not returned within their comfort zones, the compromise behavior continues. However it has to be always on the top of the highest action involved in as it is shown in Figure 7.11.

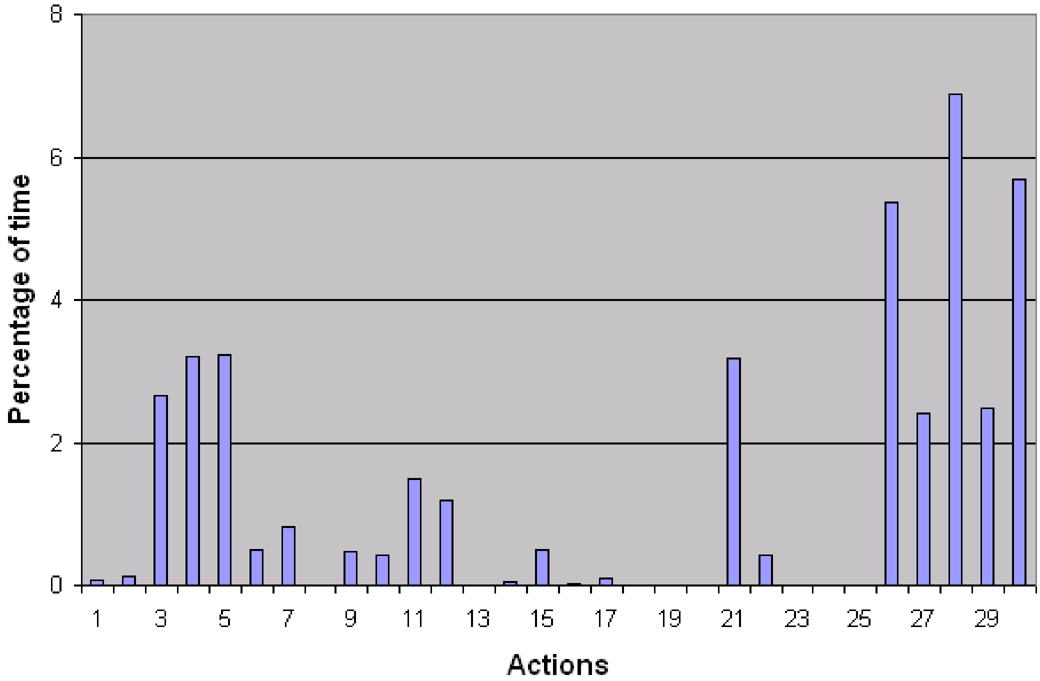

Figure 7.9: Duration of the action execution over the 65000 iterations

When compromise behaviors are available, the action selection model prefers to choose them instead of the corresponding actions sequentially (see Figure 7.9 and Table 7.3 for the correspondence between the numbers and the actions and Table 7.2 for the compromise behaviors). The compromise actions 26, 27, 28, 29, and 30 are more often chosen during the test simulation with all the model functionalities.

7.2.5 Dynamic resources

The environment has to be dynamic, i.e., the quantities (subjective or physical) of the resources should be variable during the time and the situation. They are defined in the rules and can be changed during the simulation thanks to the dynamic tab. For instance, the food quantity on the table can be set and evolve according to the hunger of the virtual human. A more subjective quantity corresponds to the lack of washing product or the life time of the phone. The decisions of the virtual human should be taken according to the quantity of the resource. If in one location the quantity is low, the system should choose another place where there are more resources if it is possible.

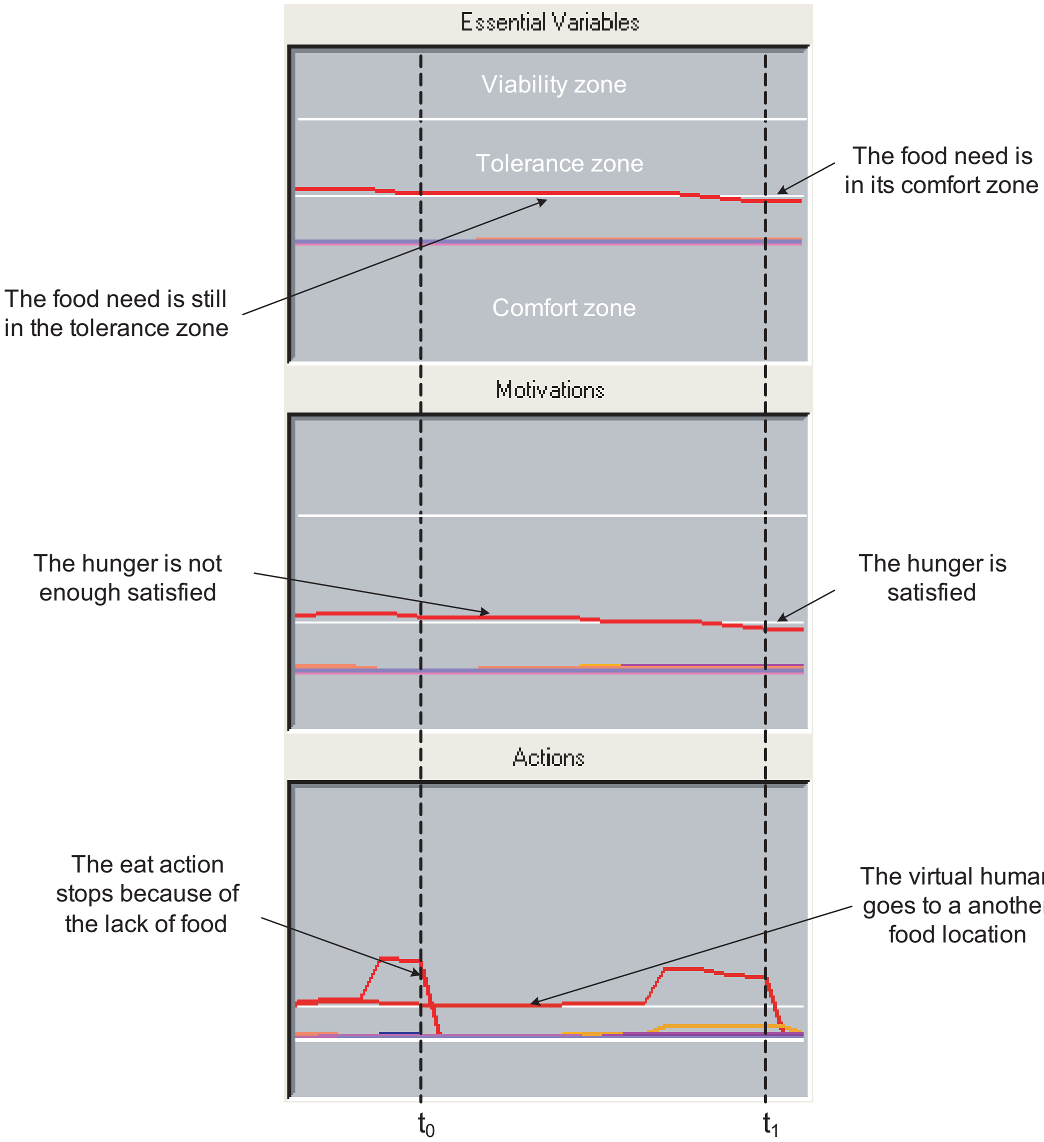

Figure 7.10: The decision-making system deals with the quantities in resource locations

Figure 7.10 illustrates an extreme case for describing the model functionalities in managing dynamic resources in real-time for clarity purposes. In this case, the virtual human has a nutritional behavior in order to satisfy his hunger. However, the quantity of food is not sufficient in the resource location where he stands. At t_0, when there is no food left, the "eat" action is stopped and the hunger is not diminishing anymore. The food need is still in the tolerance zone. So, the virtual human goes to another food location to satisfy his hunger until the food need decreases inside its comfort zone. In the second location, the "eat" action is also stopped at t_1, because of the lack of food. That is why the food need is not deeper inside the comfort zone.

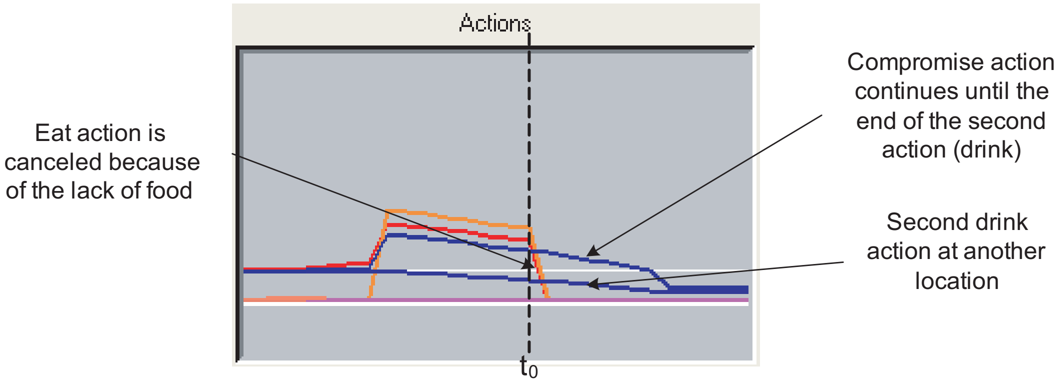

Figure 7.11: Compromise behaviors and dynamic ressources

In Figure 7.11, the virtual human performs a compromise behavior in order to satisfy his hunger and his thirst. However, at t_0, there is no food left in the resource location and the compromise action "eat and drink" is stopped. Then, since there is still water in the location the "drink" action is activated to continue decreasing the water need until it is inside its comfort zone. Once the thirst is satisfied, the virtual human should go to another food location to satisfy his hunger.

7.2.6 Urgent situations

The environment should have unexpected situations which surprise the virtual human or even put him in danger. In this case, the subjective evaluation of the motivation will give the control to the motivation that optimizes the virtual human's survival. It has to be defined by the users but by default it corresponds to avoid the danger, i.e., run away. The danger can be set at a specific point in the environment thanks to the dynamic tab (see Figure 7.12) and when the virtual human perceives it, he flees outside. The danger point can be changed during the simulation and when the user stops it, the virtual human should resume his activities. In other systems, the emotional level will achieve this [Cañamero 97].

In Figure 7.13, the virtual human performs the cook action in order to satisfy the cooking motivation. A fire suddenly starts in the oven. When the virtual human perceives

Figure 7.12: Defining the danger location easily

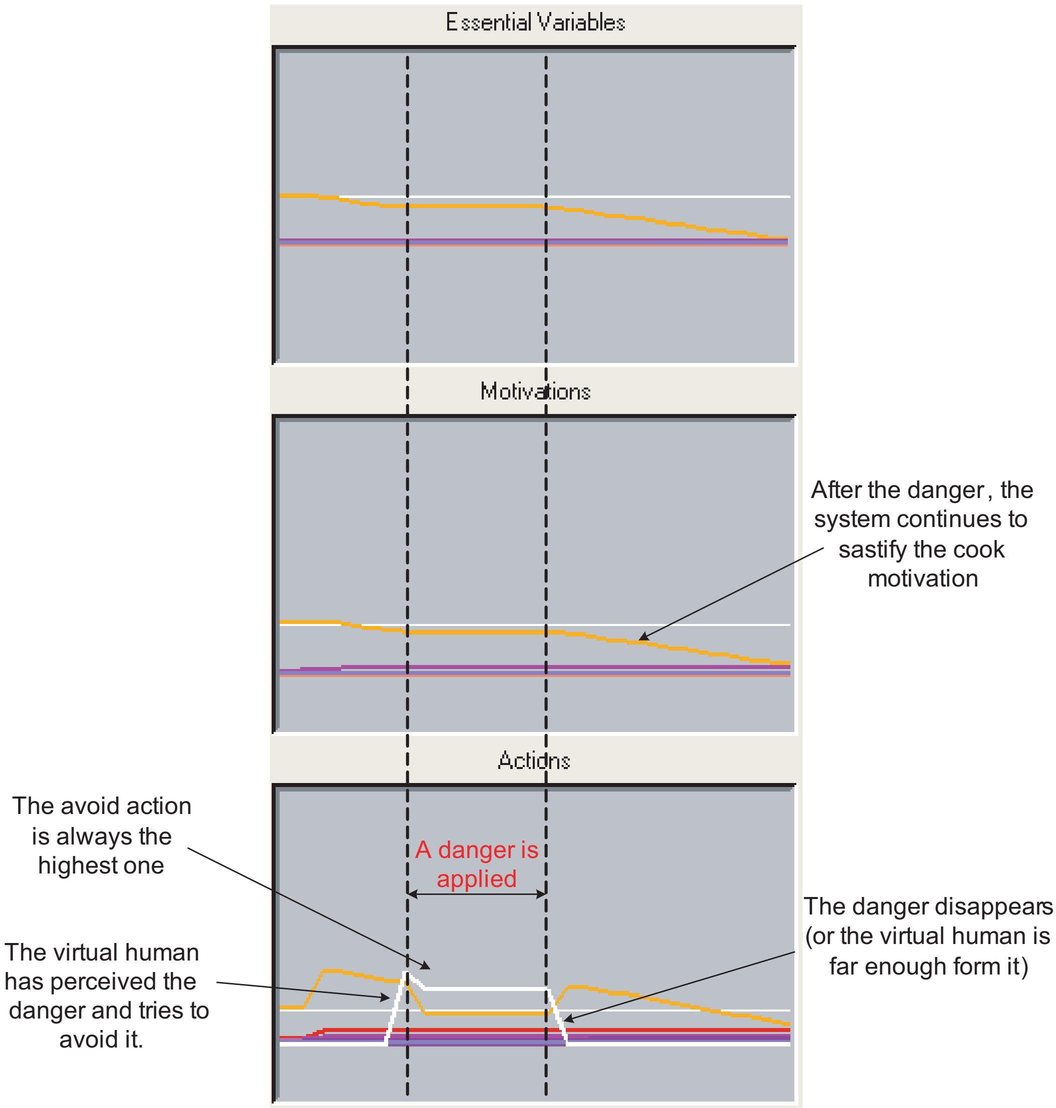

Figure 7.13: Managing danger situations

it, the "avoid" action is chosen by the action selection mechanism to ensure his survival until the danger disappears or it is far enough from him. In this king of situation, the "avoid" action is always the chosen action (see section 4.3.1). For this particular simulation, we apply the danger just a few minutes to show what happens in urgent situations: the virtual human stops his current behavior and goes to the door (the default action in danger situations). When the danger disappears, the virtual human resumes the previous behavior which was cooking.

7.3 Robust and goal-oriented architecture

Virtual humans must have complex behaviors to be believable. In addition to the fact that they need to be situated, i.e., have reactive and opportunist behaviors, the decision-making system should have a behavioral planner to execute sequences of actions. However, it should be possible to interrupt these sequences at every moment in time to satisfy another more urgent motivation. This section shows more in details the coherence of the model thanks to the behavioral planner, which allows to chain actions into plans in order to satisfy motivations. The persistence of actions allows to avoid dithering and an efficient time-sharing to consider all the motivations even if they have a low priority. Moreover chaining behaviors and considering their consequences on other behaviors are also necessary for obtaining more complex behaviors.

7.3.1 Coherence

In this environment, the system has to plan sequences of actions in order to satisfy conflicting motivations. The virtual human has to go to specific locations wherever he is. He must avoid many obstacles because the environment is narrow. Moreover, he has to achieve the intermediate actions before he can perform the motivated action to decrease the internal variable. This will test the efficiency of the path-planning module and our behavioral planner.

In the reference simulation, the coherence of the model has already been shown but not in details. Figure 7.14 shows the usual behavior of the virtual human generated by the behavioral planner (see Section 4.2.5) when he needs to satisfy one motivation (here: hunger). When the food need enters the tolerance zone (t_0), the virtual human begins to take hunger into account, thanks to a "subjective" evaluation of the motivation. Then, behavioral sequences of intermediate actions are generated ($t_0 -> t_1$), in order to

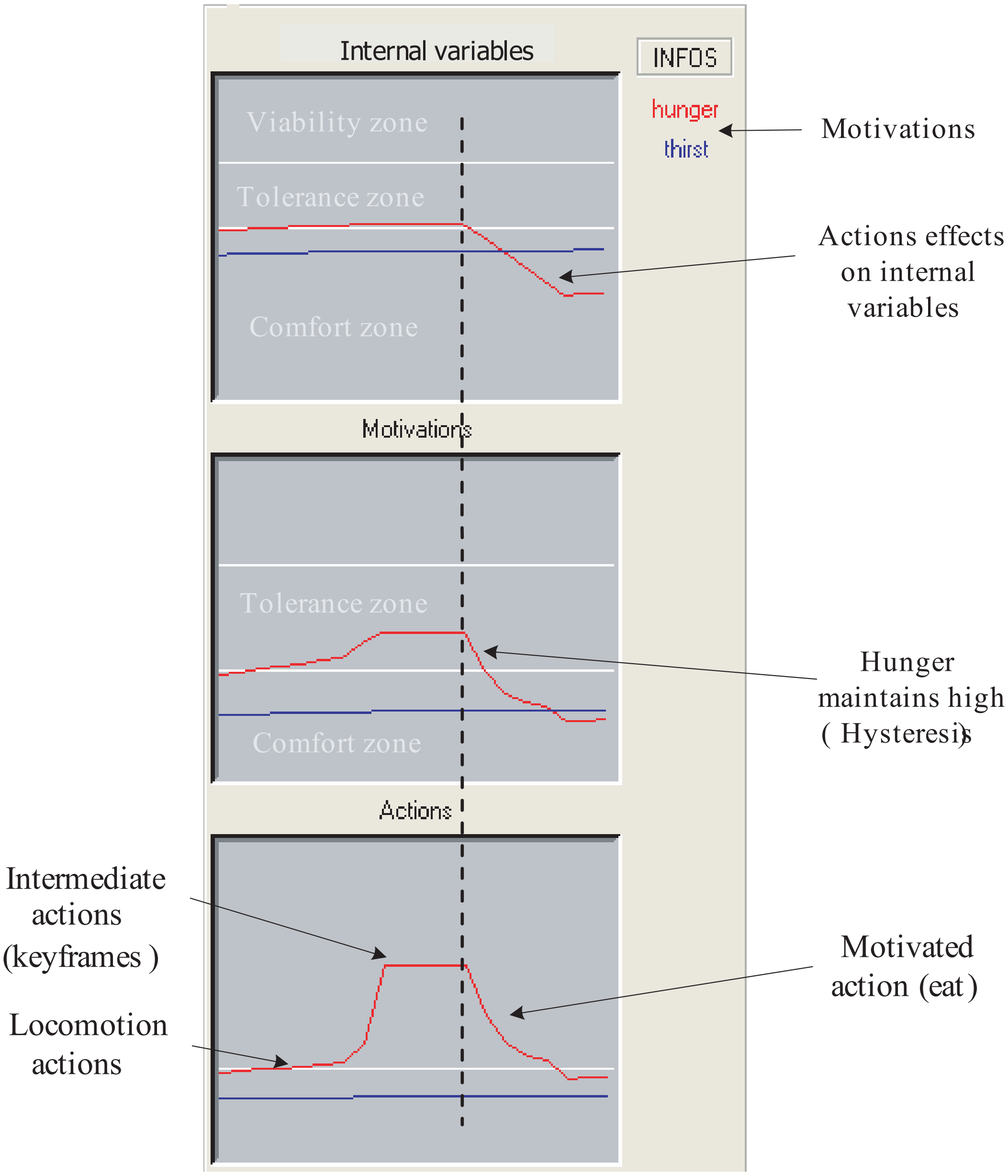

Figure 7.14: The course for satisfying one motivation (hunger) in normal conditions

reach the location where the virtual human can satisfy hunger. Finally, the eat action is performed (t_1), hunger is maintained as the highest motivation (see Section 4.1.3), and the food need decreases accordingly, until it is back within the comfort zone (t_2). Indeed the motivations are still high enough time to decrease effectively the internal variables. By default, the highest motivation drives the behavior of the virtual human and the behavioral planner finds the right sequence of actions to decrease the associated internal variable.

7.3.2 Persistence in actions

To obtain an efficient decision-making system and avoid oscillations between behaviors, the notion of persistence in actions should be considered. Indeed, if the action selection mechanism chooses the best action for the virtual human's well-being but without considering if its effect is long enough to decrease the internal variables, the virtual human may oscillate between several motivations and never satisfy one. The system has to maintain the motivated action effect as long as it takes for the internal variable to reach its comfort zone. However, if one motivation becomes very urgent, it has to stop the current behavior and try to satisfy this new motivation. The behaviors have to be coherent but flexible. For the persistence, we use the weight of the rules. Since the motivated action weights are twice the ones of intermediate rules, they have more chance to be chosen by the decision-making system. The weight difference is responsible for the sudden increase of the value in Figure 7.14. Moreover, a hysteresis (see Section 4.2.4) has been implemented to maintain the values of the motivations of the previous step. It allows to decrease the internal variables with a lesser impact on motivations. In this case, the current motivation the system is satisfying, keeps high and the associated internal variable can decrease in its comfort zone. However a threshold is set to limit the drop of the internal variable. Over a certain value, it makes no sense to continue decreasing them. The "subjective" evaluation of motivations (see Section 4.2.3) helps also to continue to choose the right action until the end of the simulation.

Thanks to this robust persistence in actions, the action selection model maintains the internal variables within the comfort zone during about 85 % of the time (see Figure 7.15) and none of the internal variables reach the danger zone. This test is made with all the functionalities, unlike the reference simulation. The percentage of internal variable presence in the tolerance zone corresponds to the time that the virtual human has focused his attention to reduce the corresponding internal variables. The differences with the reference simulation come from many factors such as the influence of the emotions, the

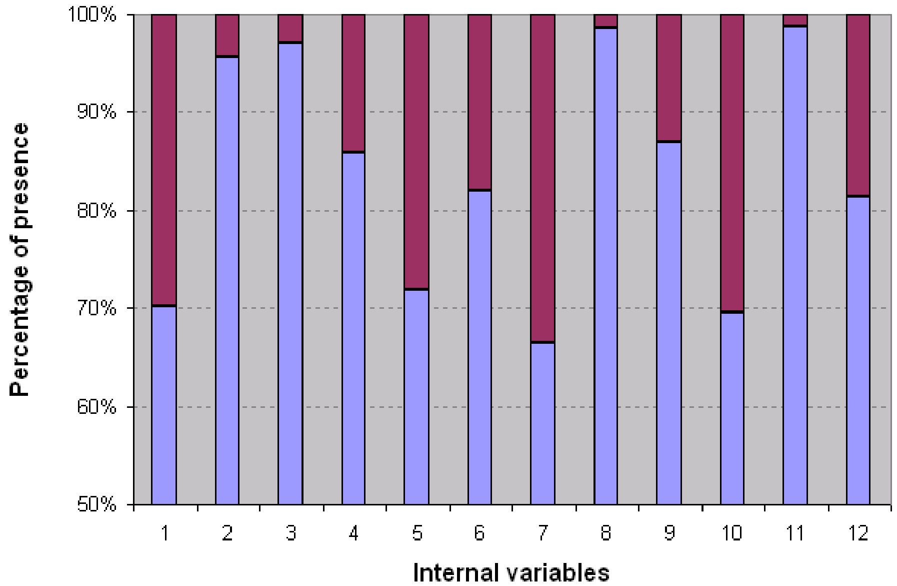

Figure 7.15: The percentage of presence for the twelve internal variables according to the threshold system: comfort (blue), Tolerance (brown) and danger (white) zones over 65000 iterations (see Table 7.1 for the corresponding internal variables to the numbers)

motivation parameters, the perceptions or the action lengths. For instance, if an internal variable exceeds the comfort threshold at the beginning of a long action, such as sleep, it will stay for a long time in the tolerance zone, because the action selection model is focused on satisfying the sleeping need. It is the opposite for the emotions: if the virtual human likes to perform an action, the model will try to satisfy this motivation as soon as possible when the associated internal variable overcomes the comfort threshold. If the virtual human doesn't like the actions, the internal variables will stay longer in the tolerance zone. It is almost the same for the motivation parameters: if they are high, the associated internal variables will often exceed the comfort threshold.

7.3.3 Time-sharing

In spite of the persistence in motivated action in order to be coherent, efficient, and to avoid oscillations, the system should also have a time-sharing notion (see Section 4.2.4). The phenomena of time-sharing in which low priority activities are given a chance to be executed despite the presence of a higher priority activity reduces the chances of pursuing a single goal to the detriment of all others [Blumberg 94]. The motivations with low priority should also be considered by the action selection model. There are twelve conflicting motivations with three types of priorities (see Section 4.1.3) and they evolve differently. To have realistic behaviors, the motivation "watering plants" should be satisfied even if it is not so important compared to hunger or thirst.

This test simulation shows the time-sharing of the virtual human according to the locations. It also corresponds to the time-sharing of the actions executed by the virtual human. In this test, all the model functionalities are activated and the virtual human has the choice between several locations to satisfy his motivations. Many other factors are also influencing the decision of what to do next. So, the repartition between the chosen locations is hardly even compared to the reference simulation. During the 65000 iterations, the virtual human has generally gone to the goals which are nearer and where he can satisfy several motivations at the same time (see Figure 7.16 and Table 7.3 for the correspondence between the numbers and the locations).

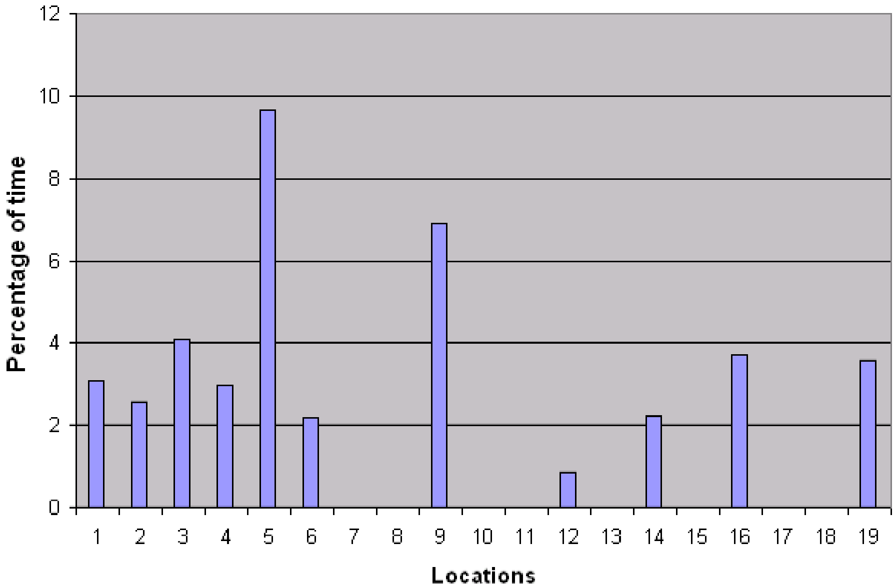

Figure 7.16: Time-sharing for the 19 goals (see Table 7.3) over 65000 iterations

Figure 7.16 represents the percentage of presence of the virtual human at the different locations. As one can observe, he stays for a while at the locations 5 and 9, because it corresponds to the sleep and the read actions which are long actions. As the action selection model prefers compromise behaviors, the virtual human is often at locations where he can perform one (1, 2, 5, and 6). This explains why he never goes to the location 7 (oven) where he can only cook instead of location 2 (sink) where he can eat, drink, and cook. The virtual human does not go to the locations 8, 10, 11, 13, 15, 17, and

(1)	food_need	hunger	table	(1)	eat	(1)
			sink		eat1	(2)
(2)	water_need	thirst	sink	(2)	drink	(3)
			table		drink1	(4)
(3)	rest_need	rest	sofa	(3)	sit	(5)
			bedroom		sit1	(6)
(4)	toilet_need	toilet	toilet	(4)	satisfy	(7)
(5)	sleep_need	sleeping	bedroom	(5)	sleep	(8)
(6)	wash_need	washing	bathroom	(6)	wash	(9)
(7)	cook_need	cooking	oven	(7)	cook	(10)
			sink		cook1	(11)
(8)	clean_need	cleaning	worktop	(8)	clean	(12)
			bathroom		clean1	(13)
			shelf	(13)	clean2	(14)
			table1	(14)	clean3	(15)
(9)	read_need	reading	bookshelf	(9)	read	(16)
			computer		read1	(17)
(10)	social_need	communicate	computer	(10)	write email	(18)
			desk1	(15)	phone	(19)
			living	(16)	phone2	(20)
(11)	sport_need	exercise	room	(11)	do push-ups1	(21)
			desk	(17)	do push-ups2	(22)
			kitchen	(18)	do push-ups3	(23)
			living room	(19)	balance	(24)
(12)	waterplant_need	watering	plant	(12)	water	(25)
		compromises	table		eat and drink	(26)
			sink		eat, drink and cook	(27)
			computer		read and play	(28)
			bedroom		sleep and rest	(29)
			bathroom		wash and clean	(30)

Table 7.3: Visualization of the hierarchy of the time-sharing test

18 because the locations 14, 16 and 19 are in the same room that the default action where the virtual human is most of his time. However if an action, such as watering plants, is far and cannot be done at another place, the action selection model chooses it whatever the distance. In this simulation test, the water action has lowest motivation and emotion parameters.

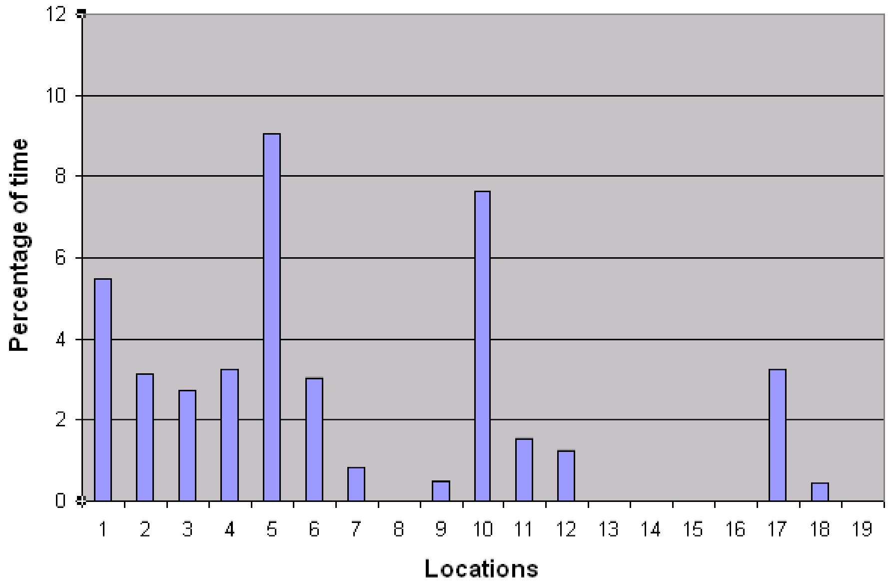

Figure 7.17: Time-sharing for the 19 locations(see Table 7.3) over 65000 iterations

In Figure 7.17, the nearest locations, i.e., 14, 16 and 19 are suppressed. In this case, the repartition of the places where the virtual human goes is different: the compromise behavior "read and play at the computer" which was farer before, is now often chosen and the locations for the "exercise" motivation are not only in the living room (locations 11, 17, and 18) because the virtual human moves more in the apartment and therefore has more opportunist behaviors. When the "exercise" need exceeds its comfort threshold, the location choice depends where the nearest one for exercising is. However, some locations which are far and don't have the ability of comprise behaviors such as 8 and 15 are never chosen in both simulations. They could be chosen if the virtual human was nearer these locations and if the associated motivation is the next to satisfy. The sum of the percentage of the virtual human presence in different locations for satisfying each motivation corresponds with the parameters defined at the beginning of the simulation. Even though some locations are not chosen, the internal variables stay about 85% of the time in the comfort zone (see Figure 7.15).

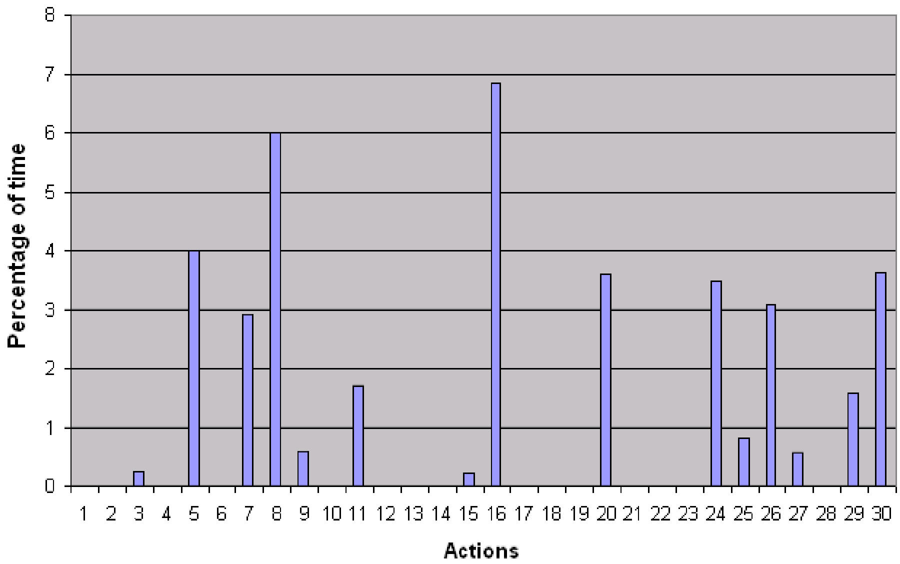

Figure 7.18: Time-sharing for the 30 actions (see Table 7.3) over 65000 iterations

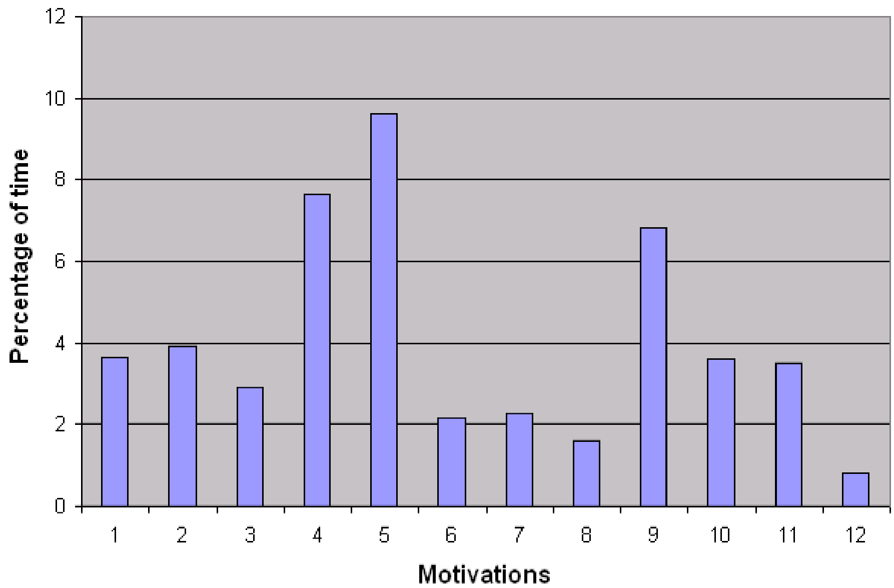

Figure 7.19: Time-sharing for the twelve motivations (see Table 7.3) over 65000 iterations

Figure 7.18 shows what does the virtual human during the simulation. Many actions are not chosen because the action selection system performs in priority compromise behaviors and nearest actions. Actions involve in compromises behaviors are rarely chosen individually. For instance, the actions 12, 13, 14, and 15 or 1, and 2 are never performed whereas the compromise actions 26, 27, and 30 are often chosen. It is the same for the actions which are far from the default action (watching the television in the sofa) where the virtual human is most of his time. The action length plays also a role in the percentage of time the virtual human performed the actions such as the "sleep" action (8). Finally many influences modulate the choices of the virtual human.

However, it does not change the overall satisfaction for each motivation depending on the configuration of the parameters (see Figure 7.19). In this test, the "cleaning" motivation parameter was the highest and the associated emotion parameter was also high. This explained why the "clean" action is the most chosen one. The motivations are satisfied in the same way than the reference simulation with the exception that the action selection system has more possible choices.

7.3.4 Behavior interactions

Chaining behaviors and considering their consequences on other behaviors is important for increasing the complexity of the scenario and the believability of the virtual human. However, it should be done after chaining motivated actions. For instance, if the virtual human does push-ups, he will get tired and the "rest" motivation should be chosen. The users can define interactions between behaviors, in this case, between doing sports and resting. The action value (rest) is increased until the virtual human executes it and then have more chance to be chosen by the action selection model. The number of interactions is not limited. Moreover, the users can define the strength of the interaction due to the influence factor in a range between 0 (no interaction) and 100% (strong interaction). By defining many interactions, the scenario can be really complex and thus the virtual human's behaviors are more realistic.

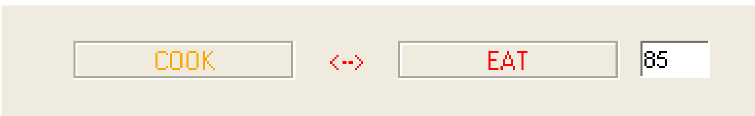

Table 7.4: Visualization of the interaction parameters

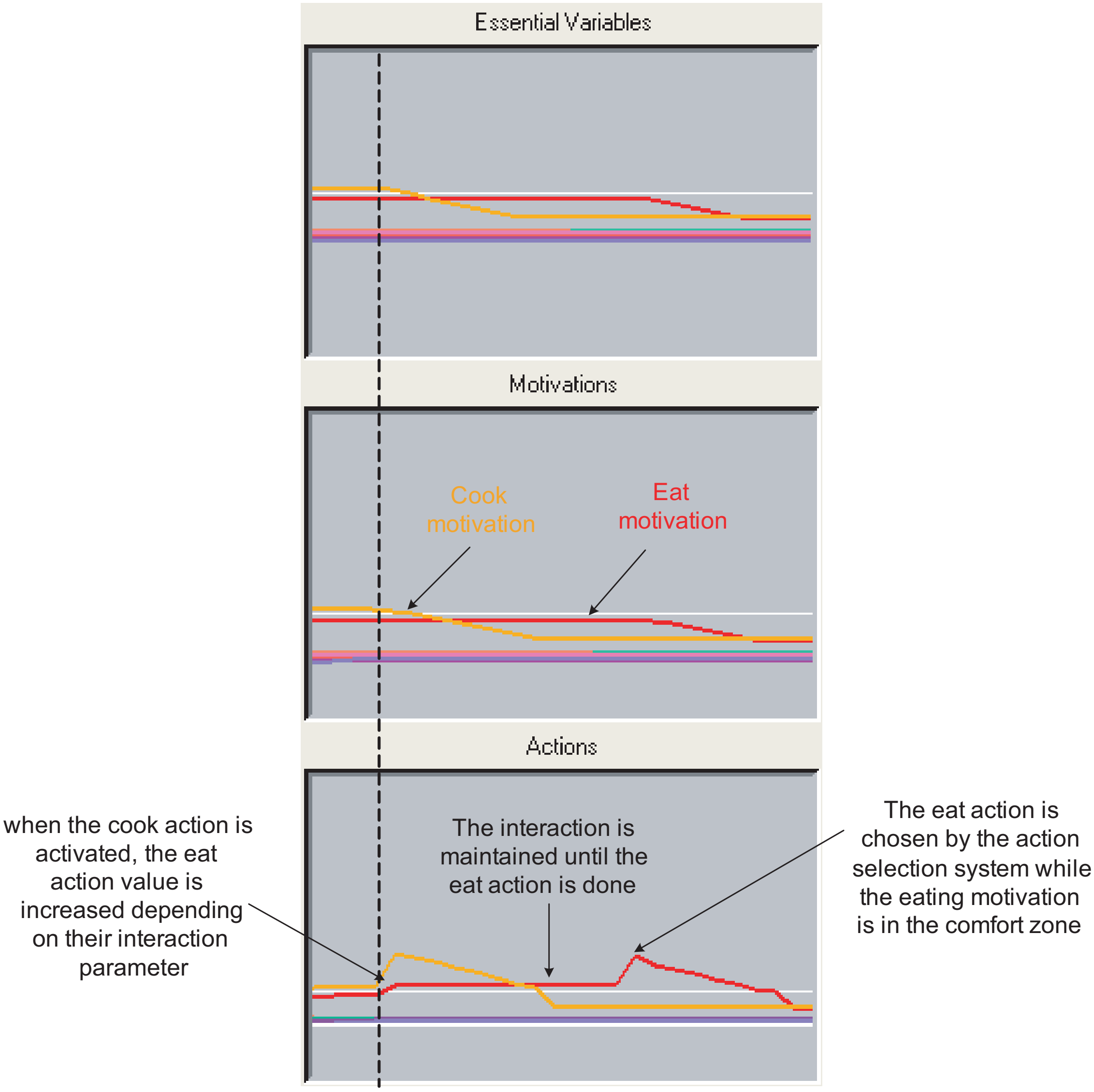

Figure 7.20: Interaction between the eating and the cooking motivations

In Figure 7.20, an interaction is defined between the "hunger" and the "cooking" motivation, i.e., after cooking the virtual human has a tendency to eat. Indeed the interaction between the two motivations is set to 85% (see Table 7.4). Note that by default there is no interaction corresponding to 0%. When the virtual human begins to cook, the eat action value is increased until this action is executed. The virtual human is performing the "cook" action. Thanks to the interaction, the "eat" action exceeds its activation threshold while "hunger" is still in its comfort zone. The "eat" action will be chosen by the action selection mechanism after the cook action is finished because the interaction is maintained until the "eat" action is done. The interactions are defined in the initialization file or during the simulation in the interaction tab. Chaining behaviors is then possible, and the flexibility of the system is maintained because these behaviors can be interrupted by another more urgent one.

7.4 Subjective architecture

To have less predictable behaviors, the action selection model integrates some functionalities that influence the decision-making depending on how the user wants the virtual human to behave. These influences are the action length, the evolution of the parameters, and the emotions. They can be changed in real-time thanks to the graphical interface. With these influences, the user can define a certain personality in order to obtain more interesting and believable virtual humans.

7.4.1 Action length

The action duration should be different depending on their type. For example, sleeping cannot last only 5 minutes. In this case, we talk about resting. In the rules or through the graphical interface, the users can change the length of the actions so that they correspond to what they want and to offer more realism and believability. To be more realistic, the user can define the action length in minutes as it is depicted in Table 7.5.

Figure 7.21 shows that the presence of the virtual human in different locations corresponds to the durations defined. It is also possible to tune the parameters to obtain a 24 hour simulation where the virtual human sleeps during the night. If the action lasts too long, the simulation can be accelerated thanks to the graphical interface. Finally this parameterization reduces the number of times that each action is chosen but does not change the time-sharing (See Figure 7.22) and the coherence of the action selection.

Actions	Length	Actions	Length	Actions	Length
Eat	15	Sleep	200	Read	30
Drink	15	Wash	30	Write email	20
Sit	25	Cook	45	Do push-up	30
Satisfy	15	Clean	25	Water	15

Table 7.5: The different action length expressed in minutes

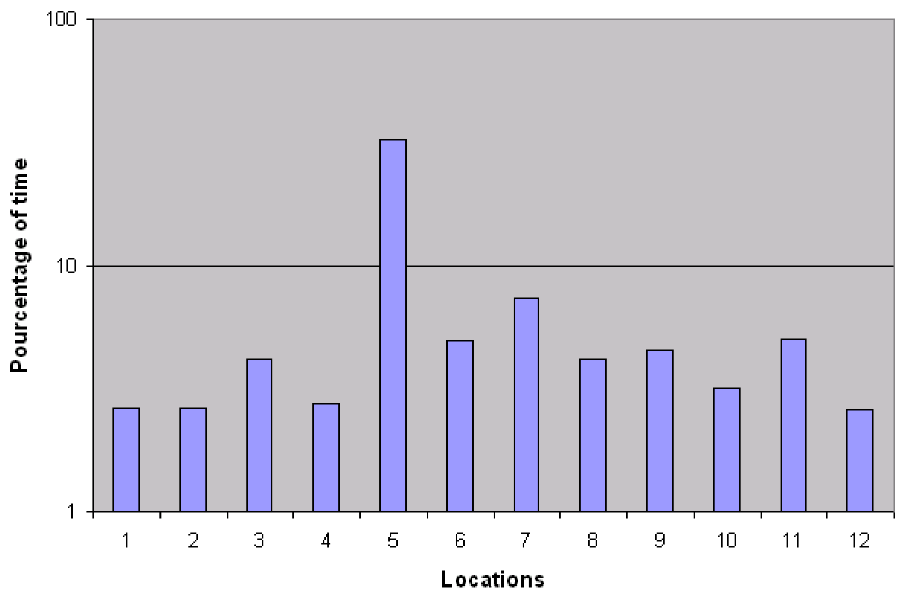

Figure 7.21: Percentage of time the virtual human is in the different locations (see Table 7.1 for the corresponding between numbers and locations)

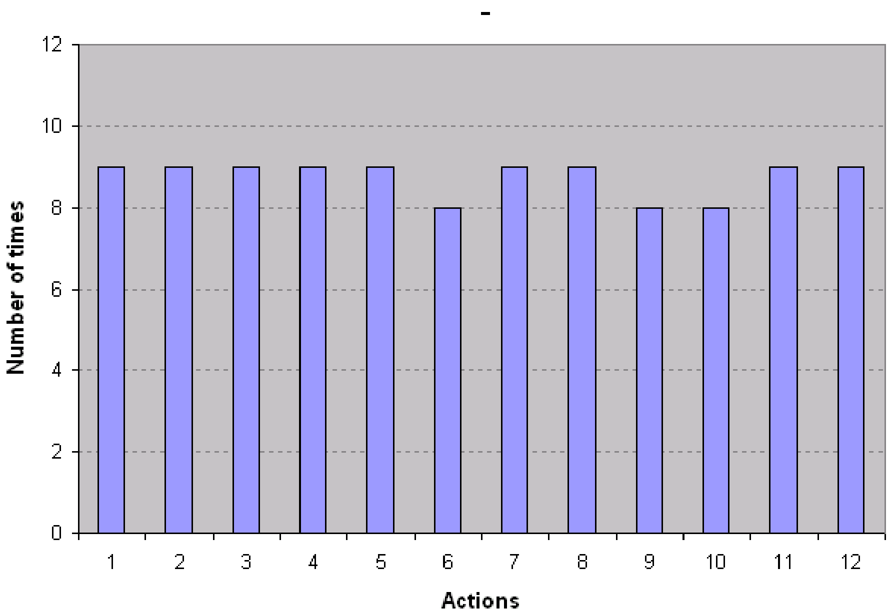

Figure 7.22: Number of times that each action is chosen by the action selection mechanism over 65000 iterations

7.4.2 Personality

By defining motivation parameters, the user can set a certain personality such as lazy, greedy, sporty, tidy, dirty, etc. to the virtual human (see Section 4.1.3). It changes the evolution speed of the internal variables associated to the motivations but is limited in a certain range (with 50% the default value). So, if a motivation is set to 100%, the associated internal variable will increase faster and the model has to satisfy this motivation more often.

Table 7.6 shows an example of tuning these parameters. In this case, the virtual human is greedy, lazy, and dirty. Figure 7.23 shows that the action selection model chooses the actions accordingly. However, the internal variables stay most of the time inside their comfort zone (see Figure 7.24).

Table 7.7 shows another example of parameters tuning (all other parameters are as in the reference simulation). In this case, the virtual human is tidy, social, and sporty. The corresponding results (see Figure 7.25) show that the virtual human chooses more often to do exercise, communicate or wash and clean. However some differences with the tuning can be observed because of opportunistic behaviors. This is an easy way to define personalities. It can be done at the beginning of the simulation and be changed in real-time during the simulation. By default, it is defined randomly.

Table 7.6: Tuning of the motivational parameters inside a certain range(50% is the default value). In this case, the virtual human is greedy, lazy, and dirty

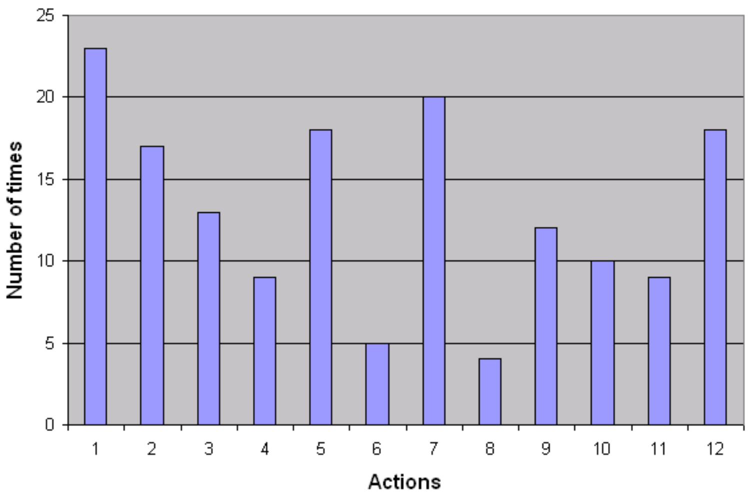

Figure 7.23: Number of times that each action is chosen by the action selection model during the 65000 iterations

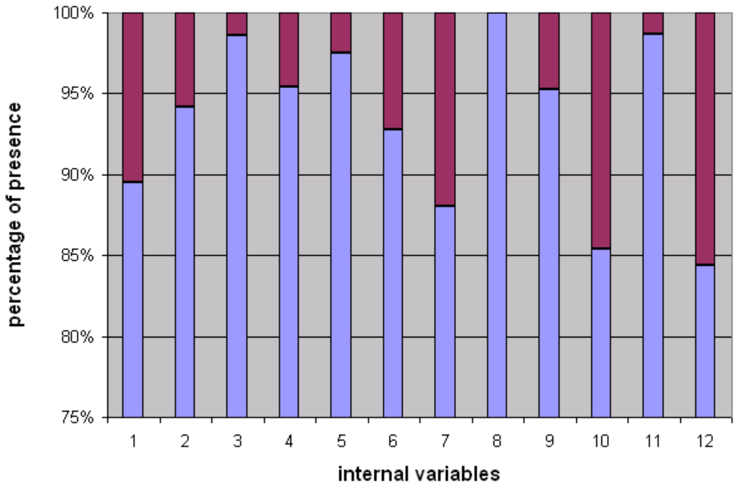

Figure 7.24: Percentage of internal variable presence according to the threshold system: comfort (blue), tolerance (brown) and viability (white) zones

Table 7.7: Another tuning of the motivation parameters inside a certain range(50% is the default value). In this case, the virtual human is tidy, social, and sporty

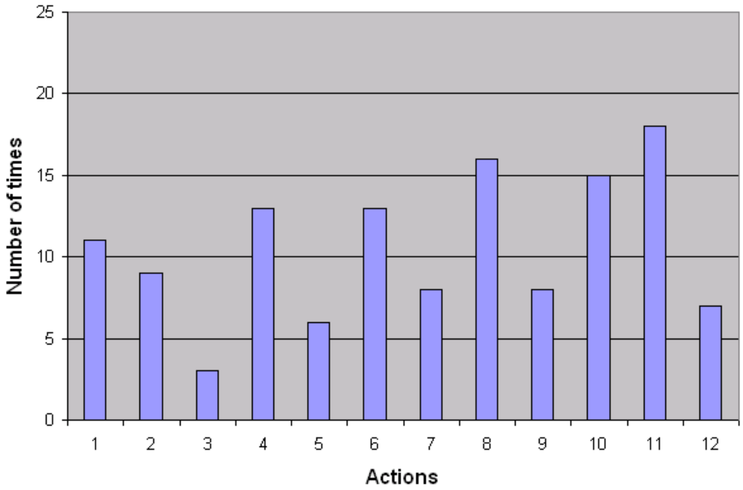

Figure 7.25: Number of times that each action is chosen by the action selection model

7.4.3 Emotions

Modulations in the motivations give the virtual human a certain personality. For instance, if we define as high the sleeping and resting motivations, the virtual human will have a sleepy behavior, i.e., he will often sleep and rest over the time. This is a first step towards personality and allows the virtual human to have a specific behavior during the simulation. However, emotions are the main way of expressing individuality and personality (see Section 4.1.4). They influence the choice by taking into account the preferences of the virtual human defined by the user. For example, if the virtual human likes sleeping, he will have a strong preference to choose this action over the others and perform it. In our system, the motivational and emotional parameters can be defined in the rules at the initialization, or with the graphical interface during the simulation. In this case, the personality of the virtual human can change in real-time. He can change his sleepy personality to a sporty one by increasing the "like" emotions for the sport actions.

For now, the emotions in our model are very simple and only determine whether the virtual human likes or dislikes to satisfy the motivations. However, they give an idea on how our model could be enhanced with more complex emotions. The like/dislike emotions are set randomly at initialization and can be changed during the simulation.

Table 7.8 shows how the emotions are set and Figure 7.26 illustrates their consequences on the number of times that each action is chosen by the action selection mechanism. As expected the more positive the emotion is, the more the decision-making system chooses

Table 7.8: Tuning of the emotional parameters inside a certain range

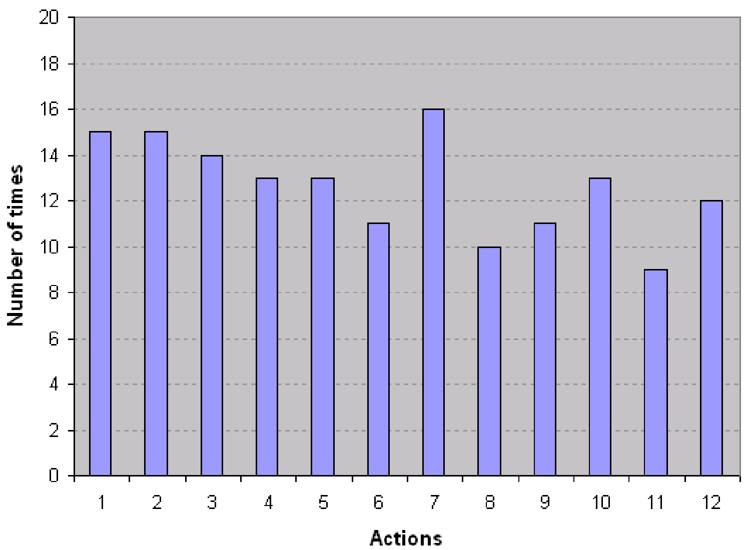

Figure 7.26: Number of times that each action is chosen during the 65000 iterations of the simulation

the associated actions and conversely. The differences between the choices are not huge because emotions just influence the decision-making.

7.5 The model generality

The action selection model is limited neither to virtual humans nor to the environment. It can be applied on any virtual creature which has some motivations. However, the motivations have to be set according to the location where the virtual creature is situated.

7.5.1 Applied to a dog

To test the generality of the system, we use it for a dog (see Figure 7.27) with a few motivations: hunger, thirst, rest and piss (see Table 7.9). As the number of motivations is not limited, we want to show with a simple example that the action selection mechanism works in all cases. We define new rules (see Appendix B) and some keyframes for the dog and apply our action selection mechanism to it.

Figure 7.27: Dog used for testing our model

The dog has a coherent and flexible behavior. Indeed the internal variables stay about 95 % of the time in their comfort zone (see Figure 7.28). This seems normal because there

motivations	goals	actions
hunger	kitchen	eat
thirst	kitchen	drink
rest	carpet living	sit
	carpet room	lie down
call for nature	door	go out

Table 7.9: Visualisation of the dog hierarchy for the test simulation

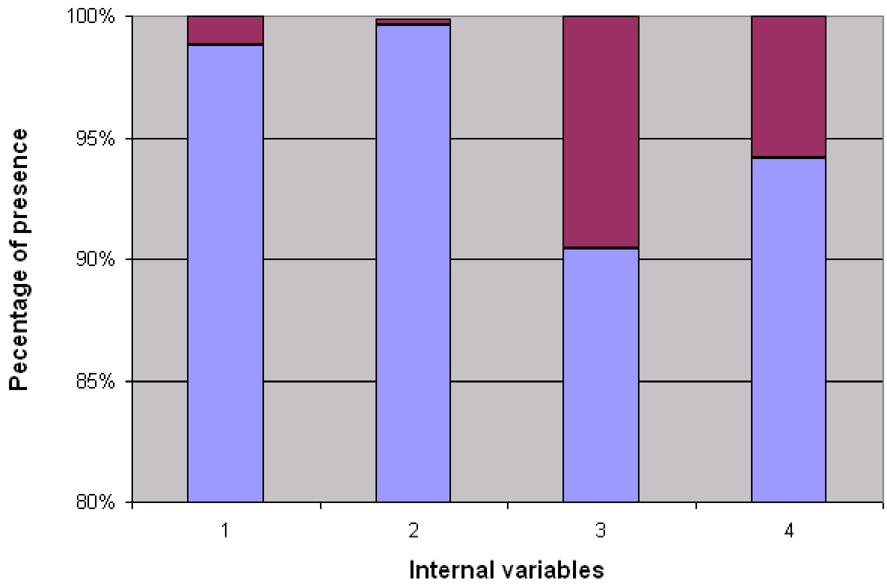

Figure 7.28: Percentage of presence of internal variables into the comfort zone.

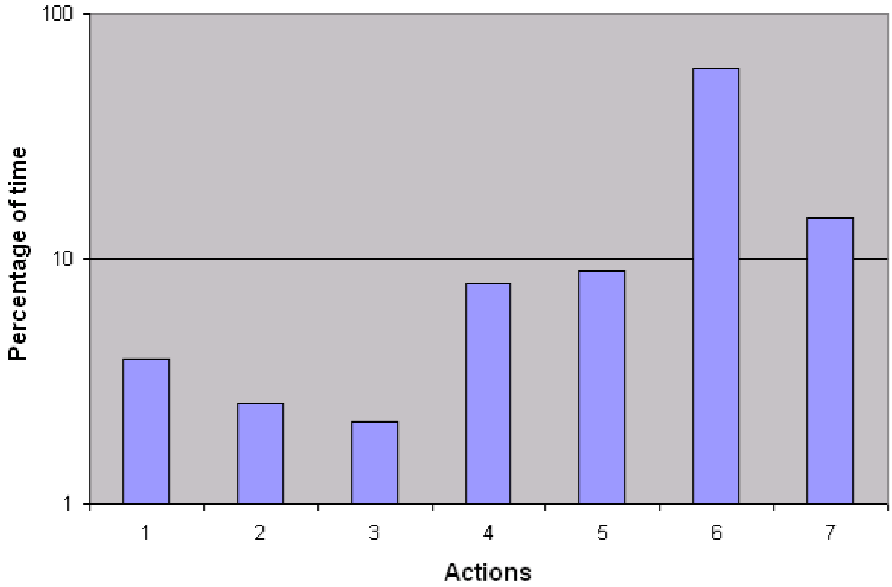

Figure 7.29: Percentage of time that each actions have been executed (see Table 7.9 for the correspondence of the numbers). The fifth action corresponds to the default one (stays where it is) and the sixth to the moving state.

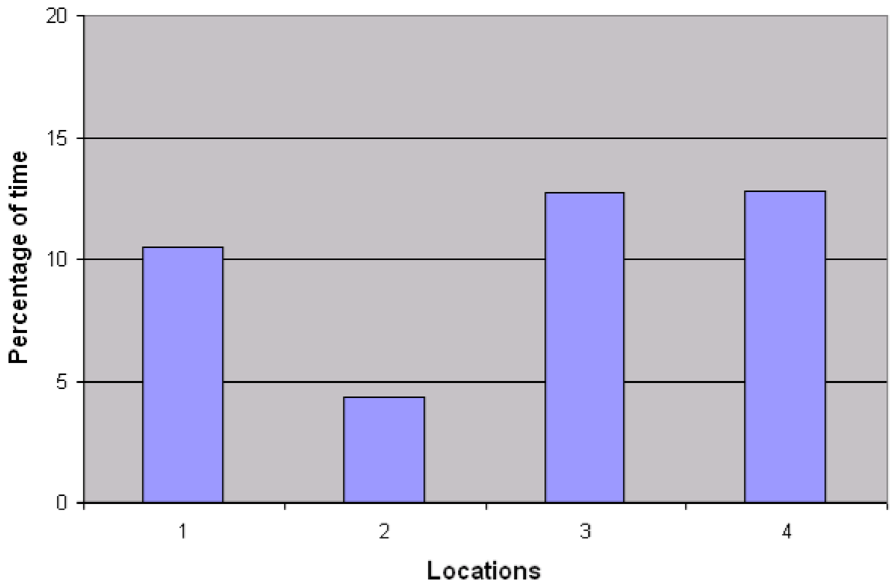

Figure 7.30: Percentage of dog presence in different locations

are very few conflicting motivations, and few influences on the decision-making system. The dog shares efficiently its time between all the possible actions according to the defined parameters (see Figure 7.29). Finally it spends a lot of time performing the default action which is "stays at the place where it is". The dog also visits the locations according to the opportunities it has. Indeed, it rests at the two possible locations where it can satisfy the "rest" motivation instead of going at only one of them (see Figure 7.30). However, it is possible to complexify the dog hierarchy as well as of the virtual human's one in order to obtain more interesting and believable behaviors.

7.5.2 In a supermarket

This simulation test shows that we can use the model in other environments. We choose a supermarket where the virtual human can have several non trivial and conflicting motivations (except the social ones) such as "buy some fruits, hygiene products, cheese, essential products, tinned food, ice cream, or multimedia products". We defined some locations for the motivated actions in the rule base system. The virtual human has a shopping list but, according to his desire or the price opportunities, he can buy something else. Some products can be out of stock and the virtual human can like some more than the others. This allows to define a personality such as thrifty or extravagant for the virtual human.

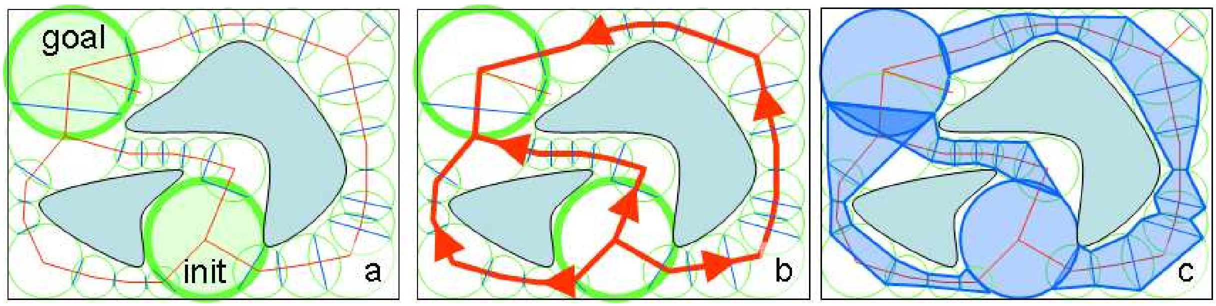

Figure 7.31: View of paths generated from the 3DSmax file

The path-planning configuration is automatically obtained from the 3DSmax file [Pettre 05] (see Figure 7.31). This path-planner is currently developed in the VRLab. It is very efficient and manages path-planning for crowds. Thanks to the walk engine [Boulic 04], a virtual human can move wherever he wants and avoids obstacles.

In Figure 7.32, the virtual human is taking sugar to satisfy the motivation "buy sugar" which is the highest one at this moment. In the end, the model is dependent on the environment only for four purposes:

Figure 7.32: Simulation of the action selection model in a supermarket

- the locations where the virtual human can satisfy his motivations,
- the path-planning,
- the danger zone,
- the names in the hierarchy often link with it.

These dependencies are contextual and the model always has to be adapted to the environment. However, its functioning and its main role, which is to maintain internal variables inside their comfort zones don't change.

7.6 Summary

Our action selection model for virtual humans in persistent worlds is flexible and reactive enough to adapt its decision-making to the environmental changes, and also robust and coherent enough to chain actions or behaviors in order to satisfy motivations. The results show that each functionality complexifies the choices of the most appropriate action at each iteration to obtain more interesting and believable behaviors instead of predictable ones. The user can set the functionality parameters and give a certain personality to the autonomous virtual human who changes his behavior accordingly. In the end, the virtual human lives his own life in his environment according to the internal and external factors. However, the user can influence his decisions at several levels by modifying different parameters in real time, thanks to the graphical interface.

Chapter 8

Conclusion

This thesis consists in designing an action selection model for autonomous virtual humans that take their decisions continuously in real-time, according to internal and external factors. Indeed, virtual humans in persistent worlds have to keep on choosing what to do next even after they have finished the specific task requested by the user. Our model is first implemented for one virtual human in a bottom-up approach, i.e., increasing the complexity progressively. This allows to try to understand the individual action selection for obtaining more complex and realistic autonomous virtual humans before focusing on social interactions.

For designing our action selection model, we use reactive and goal-oriented hierarchical classifier systems [Donnart 94] associated with the functionalities of a free flow hierarchy [Tyrrell 93a] for the propagation of the activity. Hierarchical classifier systems allow to have a coherent and robust behavior by finding sequences of actions in order to reach goals. The free flow hierarchy brings reactivity and flexibility to the hierarchical system and is necessary to effective action selection mechanisms.

To obtain a high degree of autonomy and individuality for autonomous virtual humans, an action selection architecture needs to be:

- reactive: managing unexpected and opportunist situations,
- pro-active: allowing to satisfy motivations by generating goal-oriented behaviors to reach specific goals,
- motivational: responsible for the virtual human autonomy by giving the goals to the system,

- emotional: modulating and evaluating the system choices, necessary for social interactions

The first three levels are implemented progressively in order to understand their implications and their influences on each other. Several functionalities have been designed to help, modulate and evaluate the action selection. New motivations or actions can be added easily and their number is not limited. Since many parameters can be tuned, a graphical interface allows to change them during the simulation in order to configure the model as wanted. Indeed, the user can give a personality such as lazy, greedy, sporty, tidy, dirty, etc, to the virtual human. We have just begun to implement the emotional level with the like/dislike capabilities in order to understand how the emotions modulate the other levels. Indeed, emotional influences are less clear than the motivational ones.

A scenario is created to test all the functionalities of the action selection model for individual autonomous virtual humans in persistent worlds. A virtual human has to satisfy his twelve motivations at specific locations in a virtual apartment by choosing between thirty possible actions. Moreover, the decisions must be taken in real time which implies that the behavior planner has to be quick and efficient. Finally, the chosen actions depend on many influences and become difficult to predict implying more interesting and believable behaviors for the autonomous virtual humans.

The results demonstrate that the model is coherent and flexible enough for modeling complex autonomous virtual humans in real-time. The architecture generates dynamically reactive as well as goal-oriented behaviors. Indeed, it manages persistence of actions, time-sharing, and behavior plans. It also offers the possibility of interrupting plans, compromise and opportunist behaviors. Thanks to the additional model functionalities, the decision-making is effective and consistent during long simulation tests. The most appropriate action is chosen at each moment in time with respect to many conflicting motivations, environment perceptions, and other influences, such as emotions. Moreover, the model is generic, i.e., not specific to the virtual human or the environment. Indeed, we apply it successfully to a dog and in a supermarket.

In the end, the virtual human has a high level of autonomy and lives his own life in his apartment according to internal and external factors. However, the user can influence the virtual human behaviors by modifying many parameters via the graphical interface. Applied to computer games, the non-player characters can be more autonomous and believable when they don't interact with users or execute specific tasks necessary for the game scenario.

8.1 Contributions

8.1.1 Incrementing the complexity progressively

Beginning to design directly decision architectures for virtual humans with all their complexity is difficult, while first designing decision architectures with basic motivations and generating simple goals is much easier in a "bottom-up" approach (see Section 2.1.1). This is the point of view of many ethologists: try to understand simple animal functioning in order to understand more complex ones. In our case, at the beginning the decision architecture manages a level of complexity comparable with that of simple animals. Next, by progressively increasing the complexity of the architecture and the environment, behaviors become richer and closer to those of the humans. We follow this approach by designing first a fully reactive system and then adding pro-active, motivational, and emotional levels (see Section 2.6). This way, we can well understand the implications of each level and have more interesting and believable autonomous virtual humans.

8.1.2 Strong autonomy

Autonomy is recognizably and undeniably a critical issue in the field of intelligent agents, yet it is often ignored or simply assumed. Many agents are automatic but not autonomous; they are not independent of the control of their designers. Autonomous agents are able to generate their own goals, to select between multiple alternatives the goals to pursue, and to decide to adopt goals from others (to further their own ends) [Luck 03]. If the virtual humans are autonomous, they can make their own choices of actions according to their internal states and the external factors. However, to obtain autonomy in a "strong sense" [Luck 98], the action selection model should include motivations. The self-generation of goals by the motivations is critical in achieving autonomy [Balkenius 93]. Motivations coming from internal states of agents are often missing in computational agent based systems [Luck 98]. Motivations have to be designed specifically for each autonomous virtual human before focusing on emotions, cognition or social interactions (see Section 2.5.1). They also allow to give individuality and personality to the virtual human in order to have complex behaviors. Finally, with a motivational level implemented in the decision making model, the virtual humans (or other virtual characters such as animals and even robots) can be really autonomous and give the illusion of living their own life, increasing the believability of persistent virtual environments.

8.1.3 Continuous real-time decision-making

In persistent worlds, the decision architecture should permanently choose the next action, i.e. its work is not finished once a specific task is solved. Most of the existing architectures are efficient but have a contextual autonomy in the sense that they are designed to solve specific complex tasks (cognitive architectures), to follow scripted scenarios (virtual storytelling), or to interact with other agents (BDI architectures). Autonomous virtual humans in persistent worlds need to keep on making decisions according to their internal and external factors once that complex tasks, scripted scenarios, or social interactions are finished. In this case, virtual humans need to have a strong autonomy [Luck 01]. Indeed at certain moment in time, the user cannot control the virtual human because he is always present in the virtual environment. Therefore the virtual human has to be able to take his own decision when he is not implied in specific tasks such as interacting with users.

8.1.4 Requirements for designing individual autonomous virtual humans

We summarize some criteria necessary for modeling efficient action selection architectures for autonomous virtual humans in persistent worlds in order to respect the ethological (see Section 3.2.6) and autonomous agents (see Section 2.4.2) criteria:

- **Situatedness**

  To respond quickly to the environmental changes, e.g., opportunist behaviors

- **Pro-activeness**

  To manage autonomously the fulfillment of the goals

- **Motivations**

  To give a "strong" autonomy to virtual humans by self-generating goals

- **Emotions**

  To modulate and evaluate the choice and to enhance social interactions

With these requirements fulfilled, the virtual humans are highly autonomous and distinct. They can react differently to the same situations because the decision-making is individual and have a single personality for each virtual human. The individuality has to be modeled before considering sociality. The ultimate goal of this work is to understand the individual action selection to obtain more complex and realistic autonomous virtual humans. This can be very useful for non player characters in video games so that they "live" their own life continuously. We have implemented our model of action selection by designing each criterion progressively (see Section 4.5) in a "bottom-up" approach (see Section 2.1.1).

8.1.5 Fully tuned architecture

Since many parameters are subjective, e.g., the motivation evolution or the emotional like/dislike, and have to be defined by the user, a graphical interface allows to tune them during the simulation. They have a default value, but it is subjective. A fully automatic action selection architecture is not possible in our case because too many parameters need a subjective tuning according to the environment, the users and the goal of the simulation. With a fully tunable architecture (see Section 5.3), one can configure the model as one wants in order to obtain the desired behaviors. It also allows to give a personality such as lazy, greedy, sporty, tidy, dirty, etc, to the virtual human. thanks to the motivation and emotion parameters. So, the user can define the behavior of the virtual human according to the situation. For instance, a psychologist can set the "cleaning" motivation very high and show the simulation to the patients who have such a problem.

8.1.6 Generic approach

The model is generic and is not limited in the number of motivations or actions. It can be used for any situations requiring an action selection mechanism with a "strong" autonomy. The results obtained with the dog and in the supermarket show that the model is not dependent on the actor or the environment. The architecture can even run without any graphical rendering. However the choice of actor and environment has many consequences on the model. Indeed, the virtual human takes more time to satisfy his motivations, because his environment and his animations are more complex than those of the dog. As the model is fully tunable, some adjustments are necessary depending on the environment and the actors. For instance, if the scenario takes place inside an office, the motivations are not the same than in a sport club. In a classroom, if the virtual human is a professor, the motivations are not the same as the ones of his students. The model can also be used in other domains such as robotics or economic agents on internet (see Section 8.2.2). In robotics, our model can be very useful because many constraints, non-existent in the virtual world, appear in the real one. For instance, the batteries decrease rapidly and if one motivation is defined as keeping the level of energy high, the robots will integrate it in their decision-making in order to maintain their batteries full.

8.2 Perspectives

8.2.1 Improvements

The results show that our model of action selection for autonomous virtual humans in persistent worlds is efficient and generic. However, some aspects of our model can be improved.

First, the animation part can be improved, the path-planning module [Kallmann 01] works fine with one virtual human but cannot manage many of them. A new path-planner is currently developed in the VRLab to increase this number [Pettre 05] and is used for the test in the supermarket (see Section 7.5.2). This path-planner is also easier to use because only one file has to be exported directly from the 3D design software and loaded in the simulation engine VHD++ [Ponder 03]. The integration of the smart-objects developed in the VRLab [Kallmann 01, Kallmann 03a] can also improve the simulation. With the smart objects, many possibilities of interactions between the virtual humans and the objects are possible because they contain all the information needed to use them, e.g., the place where the hands should be placed to grasp the object, the possible movements that the virtual human can do with the object, etc. This allows to avoid the use of representations databases. In this case, the virtual human can directly interact with all the objects if well tuned instead of designing keyframes.

Second, the complexity of the model can be improved. In our model, the behavioral planner is simple and can be replaced by one managing complex behaviors [Ciger 05], also developed in the VRLab. This planner can solve general actions such as "buy 2 train tickets for London" and find a solution inside a rule base to accomplish it. In this case, motivations generate the goal to reach and the behavioral planner finds plans to accomplish the action. We also have to complexify the architecture by adding more motivations (the number is not limited), more functionalities to evaluate the decision-making, and more parameters tuned by the user via the graphical interface to obtain more subtle personalities. The environment has to be also more complex with more opportunistic behaviors, the zones with dangers, etc. In the end, the decision-making will be less predictable so that more interesting and believable behaviors for the autonomous virtual humans are developed.

Finally, also emotions enhance the autonomy and the individuality of the virtual humans by giving them a certain personality (see Section 2.5.2) in addition to motivations. The emotional level of our model needs to be improved because we have just begun to implement it in a "bottom-up" approach (see Section 2.1.1) in order to well understand its implications with the other levels. The principal problem when one implements it in an action selection mechanism is to know which emotions will influence which parameters in the model. The role of emotions is less clear than the one of motivations. Since an emotional system is a complex system connected to many other behavioral and cognitive subsystems, it can act on all these systems at the same time. However, "emotions contribute to the generation of richer, more varied, and flexible behaviors" [Cañamero 01]. Although in our model the subjective evaluation of motivations replaces many roles of emotions (see Table 4.1), they are necessary for evaluating the decision making and managing social interactions.

8.2.2 Concrete applications

The most obvious applications where our model can be used are non-player characters in video games. Although graphics technology allows the creation of environments looking incredibly realistic, the behavior of non-player characters often leads to a shallow and unfulfilling game experience [Namee 01]. Everyone who has played computer games has observed that characters controlled by the computer are neither very intelligent nor autonomous (even in The Sims [Sims 2 04]). For instance, in role-playing games [The Elder Scrolls 3 03, NeverWinter Night 04, World of Warcraft 05], the non-player characters inhabiting persistent virtual worlds should give the illusion of living their own life instead of staying static or having limited or scripted behaviors. With our architecture of action selection for autonomous virtual humans in persistent worlds, the non-player characters can make their own decisions according to their internal and external factors when one doesn't interact with them. When one plays with one's favorite character and walks across a town, each non-player character can have his own motivations and occupations. For instance, the baker makes his bread most of the time but sometimes he drinks a beer at the bar, sleeps, talks with his friends or listens to some music. Finally, the game will be more interesting and believable.

However our model can be used in another application. In 3D animation films, it can help to animate secondary characters avoiding to design all their behaviors. In crowd simulations, the majority of the virtual humans does the same actions but there is always one of them that needs to perform complex behaviors because he is the leader or near to the camera. In this case, our model can be applied on some virtual humans in a crowd to give them a more realistic and believable behavior. In robotics also, it can be useful for the management of the energy power. If a motivation is associated to keep high the level of energy, the robots will take it into account in their decisions. The problems of a robot are more or less similar with virtual humans with some extra-constraints due to the real world. However, all the virtual human motivations can be applied to a robot in a similar environment. Finally psychologists can also use it for therapies. They can tune the model as they want in order to obtain the right behavior from the virtual humans.

8.2.3 Future extensions

Our model of action selection for autonomous virtual humans in persistent worlds should manage in the future social interactions and storytelling.

Social interactions are naturally following the implementation and test of complex individual autonomous virtual humans. In a "bottom-up" approach (see Section 2.1.1), we begin with a single other virtual human in order to well understand the necessary communication capabilities for two virtual humans such as body and verbal language, opinions on the others, taking into account other in the decisions, management of resource conflict, etc. Indeed, if the food location is not available because another virtual human is there, many choices are then possible: wait, ask some help, find another location, etc. This work is huge but can give some very interesting results. Emotions can be the principal functionality to begin to integrate social interactions in the model. However, we should know which parameters they should influence and with which strength because it is very subjective for each virtual human. The parameters have to be tuned in real-time with the graphical interface. Finally, the emotions will give richer individualities and personalities. Social capabilities will help virtual humans to collaborate to satisfy common goals. When we have well understood the complexity of the problem with two virtual humans, we will consider group relations.

Another possible extension is storytelling. This can be very useful for following a pre-defined story but without controlling the actors all the time. In this case, the motivations and the actions are oriented in a certain direction to perform specific actions. By chaining behaviors and with a more complex behavioral planner, it is possible to add storytelling capabilities to our model. This can be useful for the non-player characters involved in the evolution of the game. Indeed, they can live their own life but maintain the coherence of the game story. However, the social interactions have to be implemented first. For more immersion in the world of the virtual humans, the user can interact with them through virtual reality devices.

Finally testing our action selection architecture on robots can be very interesting to improve it. Indeed the real world is more complex than a simulated one and many problems that we don't have in our simulation will appear when using robots. This allows to give robustness and enhance the generality of our action selection model.

Bibliography

[Agre 90] P. Agre & D. Chapman. What are plans for ? The MIT Press, maes p. edition, 1990.

[Anderson 93] J. Anderson. Rules of the mind. Lawrence Erlbaum Associates, 1993.

[Arkin 98] R.C. Arkin. An behavior-based robotics. MIT Press, 1998.

[Ashby 52] W. Ashby. Design for a brain. Chapman and Hall, 1952.

[Aylett 96] R. Aylett & A. Jones. *Planner and domain: Domain configuration for a task planner.* International Journal of Expert Systems, vol. 9, no. 2, pages 279–316, 1996.

[Badler 99] N. Badler, M. Palmer & R. Bindiganavale. *Animation control for real-time virtual humans.* In Communications of the ACM, pages 64–73, 1999.

[Baerends 76] G. Baerends. *The functional organization of behaviour.* Animal behaviour, vol. 24, pages 726– 738, 1976.

[Bailey 90] W.J. Bailey, R.J. Cunningham & L. Lebel. *Song power, spectral distribution and female phonotaxis in the bushcricket requena verticalis (tettigoniiddae: Orthoptera): active femail choice or passive attraction.* Animal Behavior, 1990.

[Baillie-de Byl 04] P. Baillie-de Byl. Programming believable characters for computer games. Charles River Media, 2004.

[Balkenius 93] C. Balkenius. *The roots of motivation.* In J.-A. Mayer, H. L. Roitblat & S. W. Wilson, editeurs, From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior. Cambridge, MA: MIT Press/Bradford Books, 1993.

[Blumberg 94] B. Blumberg. *Action-Selection in Hamsterdam: Lessons from Ethology.* In D. Cliff, P. Husbands, J.A. Meyer & S.W. Wilson, editeurs, the Third International Conference on the Simulation of Adaptative Behavior, pages 108– 117. MIT Press Cambridge, 1994.

[Blumberg 95] B. Blumberg & T. Galyean. *Multi-level Direction of Autonomous Creatures for Real-Time Virtual Environments.* In SIGGRAPH 95, 1995.

[Blumberg 96] B. Blumberg. *Old Tricks, New Dogs: Ethology and Interactive Creatures.* dissertation, 1996.

[Bogner 99] M.B. Bogner. *Realizing consciousness in software agents.* PhD thesis, 1999. Adviser-Stan Franklin.

[Boulic 04] R. Boulic, B. Ulciny & D. Thalmann. *Versatile Walk Engine.* Game Development, vol. 1, no. 1, 2004.

[Bower 82] G. Bower & P. Cohen. *Emotional Influences in Memory and Thinking: Data and Theory.* In M.C. Fiske, editeur, Affect and Cognition: The 17th Annual Carnegie Symposium on Cognition, pages 291–331. Erlbaum, 1982.

[Brooks 86] R.A. Brooks. *A Robust Layered Control System for a Mobile Robot.* Journal of Robotics and Automation, vol. RA-2, pages 14– 23, 1986.

[Brooks 91] R.A. Brooks. *Intelligence without representation.* Artifificial Intelligence, vol. 47, no. 1-3, pages 139–159, 1991.

[Bryson 00] J. Bryson. *Hierarchy and Sequence vs. Full Parallelism in Reactive Action Selection Architectures.* In Meyer J.A. Cliff D. Husbsands P. & S.W. Wilson, editeurs, The Sixth International Conference on the Simulation of Adaptive Behavior (SAB2000), 2000.

[Bryson 03] J. Bryson. *Action Selection and Individuation in Agent Based Modelling.* In D.L. Sallach & C. Macal, editeurs, Agent 2003: Challenges of Social Simulation, 2003.

[Bryson 04] J.J. Bryson. *Action Selection and Individuation in Agent.* In D.L. Macal, editeur, Agent 2003: Challenges of Social Simulation, 2004.

[Cañamero 97] L. Cañamero. *Modeling motivations and emotions as a basis for intelligent behavior.* In W.l. Johnson, editeur, Proceedings of the first international conference on Autonomous agents, pages 148–155. ACM Press, 1997.

[Cañamero 98] L. Cañamero. *Issues in the Design of Emotional Agents.* In Emotional and Intelligent: The Tangled Knot of Cognition. Papers from the 1998 AAAI Fall Symposium, pages 49–54. AAAI Press, 1998.

[Cañamero 01] L. Cañamero. *Emotions and Adaptation in Autonomous Agents: A Design Perspective.* Cybernetics and Systems: An International Journal, vol. 32, no. 5, pages 507–529, 2001.

[Camelot 02] Dark Age Of Camelot. http://www.camelot-europe.com/. Mythic Entertainment, wanadoo edition edition, 2002.

[Castelfranci 95] C. Castelfranci. *Guarantees for Autonomy in Cognitive Agent Architecture.* In Agent Theories, Languages and Languages (ATALŠ94). Springer, 1995.

[Ciger 05] J. Ciger, T. Abaci & D. Thalmann. *Planning with Smart Objects.* In The 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: WSCG ' 2005, 2005.

[Clark 97] A. Clark. Being there - putting brain, body and world together again. MIT Press/Bradford Books, 1997.

[Crabbe 02] F.L. Crabbe. *Compromise candidates in positive goal scenarios.* In ICSAB: Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats, pages 105–106. MIT Press, 2002.

[Damasio 94] A.R. Damasio. DescartesŠ error: Emotion, reason and the human brain. G. P. PutmanŠs Sons, 1994.

[Darwin 59] C. Darwin. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. John Murray, 1859.

[Dawkins 76] R. Dawkins. *Hierarchical Organization: a Candidate Principle for Ethology.* In Bateson P. & Hinde R., editeur, Growing Points in Ethology. Cambridge University Press, 1976.

[Decugis 98] V. Decugis & J. Ferber. *Action Selection in an Autonomous Agent with a Hierarchical Distributed Reactive Planning Architecture.* In K.P. Sycara & M. Wooldridge, editeurs, Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98), pages 354–361. ACM Press, 1998.

[Donnart 94] J.-Y. Donnart & J.-A. Meyer. *A Hierarchical Classifier System Implementing a Motivationally Autonomous Animat.* In D. Cliff, P. Husbands, J.-A. Meyer & S.W. Wilson, editeurs, From Animals to Animats 3. Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB94), pages 144–153. A Bradford Book. MIT Press, 1994.

[Donnart 96a] J. Donnart & J. Meyer. *Hierarchical-map Building and Self-positioning with MonaLysa.* Adaptive Behavior, vol. 5, no. 1, pages 29–74, 1996.

[Donnart 96b] J. Donnart & J. Meyer. *Learning reactive and planning rules in a motivationally autonomous animat.* Transactions on Systems, Man, and Cybernetics, vol. 26, pages 381–395, 1996.

[Donnart 96c] J.-Y. Donnart & J.-A. Meyer. *Spatial exploration, map learning, and self-positioning with MonaLysa.* In P.e.a. Maes, editeur, From animals to animats 4. Proceedings of the Fourth

International Conference on Simulation of Adaptive Behavior, pages 204–213. The MIT Press., 1996.

[Donnart 98] J.Y. Donnart. *Architecture cognitive et proprietes adaptatives d'un animat motivationnellement autonome.* Phd thesis, LIP6 AnimatLab, Universite Pierre et Marie Curie, 1998.

[Everquest2 04] Everquest2. http://everquest2.station.sony.com/. Sony Online Entertainment, ubisoft edition, 2004.

[Franklin 95] S. Franklin. Artificial minds. MIT Press, 1995.

[Franklin 97] S. Franklin & A. Graesser. *Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents.* In ECAI '96: Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages, pages 21–35. Springer-Verlag, 1997.

[Franklin 01] S. Franklin. *Building life-like ŚconsciousŠ software agents.* AI Commun., vol. 13, no. 3, pages 183–193, 2001.

[Franklin 02] S. Franklin. *'Conscious' software: a computational view of mind.* pages 1–45, 2002.

[Frijda 95] N. Frijda. Emotions in robots, pages 501– 516. The MIT Press, 1995.

[Girard 02] B. Girard, V. Cuzin, A. Guillot, K.N. Gurney & T.J. Prescott. *Comparing a brain-inspired robot action selection mechanism with Świnner-takes allŠ.* In Proceedings of the seventh internatinal conference on simulation of adaptive behavior: From animals to animats, pages 75–84, 2002.

[Gratch 02] J. Gratch, J. Rickel, E. André, J. Cassell, E. Petajan & N. Badler. *Creating Interactive Virtual Humans: Some Assembly Required.* IEEE Intelligent Systems, vol. 17, no. 4, pages 54–63, 2002.

[Gratch 04] J. Gratch & S. Marsella. *Evaluating the Modeling and Use of Emotion in Virtual Humans.* In AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, pages 320–327. IEEE Computer Society, 2004.

[Grayson 00] J. Grayson. Python and tkinter programming. 2000.

[Grefenstette 88] J. Grefenstette. *Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms.* Machine Learning, vol. 3, pages 225–245, 1988.

[Guillot 98] A. Guillot & J. Meyer. *Synthetic Animals in Synthetic Worlds.* In I.K. Luciani, editeur, Cyber Worlds, pages 144–153. Springer Verlag, 1998.

[Guillot 00] A. Guillot & J.-A. Meyer. *From SAB94 to SAB2000: What's new, Animat?* In J.-A. Meyer, Berthoz, A., Floreano, H. D.and Roitblat & S. W. Wilson, editeurs, From animals to animats 6: Proceedings of the sixth international conference on simulation of adaptive behavior, pages 3–12, 2000.

[Hawes 01] N. Hawes. *Anytime Planning For Agent Behaviour.* In Proceedings of the Twelth Workshop of the UK Planning and Scheduling Special Interest Group, pages 157–166, 2001.

[Hendriks-Jansen 96] H. Hendriks-Jansen. Catching ourselves in the act: Situated activity, interactive emergence, evolution, and human thought. MIT Press, 1996.

[Holland 75] J. Holland. *Adaptation in natural ans artificial system.* 1975.

[Holland 86] J. Holland. *Escaping brittleness: The possibilities of generalpurpose learning algorithms applied to parallel rulebased systems.* In R. Michalski & C.J. T.M., editeurs, Machine learning: An Artifcial Intelligence approach. Morgan Kaufmann, 1986.

[Humphrys 96] M. Humphrys. *Action Selection methods using Reinforcement Learning.* Phd thesis, University of Cambridge, 1996.

[Isla 02] D. Isla & B. Blumberg. *New challenges for character-based ai for games.* In AAAI Spring Symposium on AI and Interactive Entertainment, 2002.

[Kallmann 01] M. Kallmann. *Object Interaction in Real-Time Virtual Environments.* PhD thesis, Swiss Federal Institute of Technology (EPFL), 2001. thesis number 2347.

[Kallmann 03a] M. Kallmann, A. Aubel, T. Abaci & Thalmann. *Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping.* Computer Graphics Forum, vol. 22, no. 3, page 313, 2003.

[Kallmann 03b] M. Kallmann, H. Bieri & D. Thalmann. *Fully Dynamic Constrained Delaunay Triangulations.* In Mueller H. Brunnett G. Hamann B. & L.L. eds, editeurs, Geometric Modelling for Scientific Visualization, 2003.

[Kandel 95] E. Kandel, J. Schwartz & T. Jessel. Essentials of neural science and behavior. Appleton and lange, 1995.

[Laird 87] J.E. Laird, A. Newell & P.S. Rosenbloom. *SOAR: an architecture for general intelligence.* Artif. Intell., vol. 33, no. 1, pages 1–64, 1987.

[Laird 02] J. Laird. *Research in human-level AI using computer games.* Communications of the ACM, vol. 45, no. 1, 2002.

[Latimer 87] W. Latimer & M. Sippel. *Acoustic cues for female choice and male competition in the tettigonia cantans.* Animal Behavior, 1987.

[LeDoux 96] J. LeDoux. The emotional brain. Simon and Scuster, 1996.

[Lorenz 81] K. Lorenz. Foundations of ethology. Springler-Verlag, 1981.

[Luck 98] M. Luck & M. d'Inverno. *Motivated Behavior for Goal Adoption.* In C. Zhang & D. Lukose, editeurs, Multi-Agent Systems: Theories, Languages and Applications - Proceedings of the Fourth Australian Workshop on Distributed Artificial Intelligence, pages 58–73. Springer, 1998.

[Luck 01] M. Luck & M. d'Inverno. *Autonomy, A Nice Idea in Theory.* Lecture Notes in Computer Science, vol. 1986, page 351, 2001.

[Luck 03] M. Luck, S. Munroe & M. d'Inverno. *Autonomy: Variable and Generative.* In H. Hexmoor, Castelfranchi, C. & R Falcone, editeurs, Agent Autonomy, pages 9–22. Kluwer, 2003.

[Luger 94] G.F. Luger, P. Johnson, C. Stern, J. Newman & R. Yeo. Cognitive science: The science of intelligent systems. Academic Press, 1994.

[Lutz 96] M. Lutz. Programming python. O'Reilly (Bonn and Sebastopol, CA), 1996.

[Maes 90] P. Maes. *Situated agents can have goals.* In Special issue of journal of Robotics and Autonomous vehicle control. North-Holland, 1990.

[Maes 91] P. Maes. *A bottom-up mechanism for behavior selection in an artificial creature.* In J.A. Meyer & J.A. Meyer, editeurs, the First Internationnal Conference on Simulation of Adaptive Behavior. MIT Press/Bradford Books, 1991.

[Maes 92] P. Maes. Behaviorbased artifcial intelligence, pages 2–10. The MIT Press/Bradford Books, 1992.

[Maes 93] P. Maes. *Behavior-based artificial intelligence.* In Proceedings of the second international conference on From animals to animats 2 : simulation of adaptive behavior, pages 2–10. MIT Press, 1993.

[Maes 94] P. Maes. *Modeling adaptive autonomous agents.* Artificial Life, I, vol. (1 and 2), no. 9, 1994.

[Magnenat-Thalmann 04] N. Magnenat-Thalmann & D. Thalmann. Handbook of virtual humans. John Wiley, magnenat-thalmann. n edition, 2004.

[Malcolm 97] C. Malcolm. *A hybrid behavioural/knowledge-based approach to robotic assembly.* Rapport technique, University of Edinburgh, 1997.

[Maslow 54] A.H. Maslow. Motivation and personality. Harper, 1954.

[Mataric 98] M.J. Mataric, M. Williamson, J. Demiris & A. Mohan. *Behavior-based primitives for articulated control.* In Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5, pages 165–170. MIT Press, 1998.

[Maturana 75] H.R. Maturana. *The organization of the living: A theory of the living organization.* International Journal of Man-Machine Studies, vol. 7, pages 313–332, 1975.

[Maturana 80] H.R. Maturana & F.J. Varela. Autopoiesis and cognition: The realisation of the living. D. Reidel Publishing Co, boston studies in th edition, 1980.

[McFarland 74] D. McFarland. *Time-Sharing as a Behavioral Phenomenon.* Advances in Animal Behavior, vol. 5, 1974.

[McFarland 75] D. McFarland & R. Sibly. *The behavioural final common path.* Philosophical Transactions of the Royal Society (Series B), vol. 270, pages 265–293, 1975.

[McFarland 89] D. McFarland. Problems of animal behaviour. Wiley, 1989.

[McFarland 91] D. McFarland. *Defning Motivation and Cognition in Animal.* International Studies in the Philosophy of Science, vol. 5, no. 2, pages 153–170, 1991.

[McFarland 93] D. McFarland & T. Bosser. Intelligent behavior in animals and robots. The MIT Press/Bradford Books, 1993.

[Meyer 95] J.-A. Meyer. The animat approach to cognitive science., pages 27–44. The MIT Press, 1995.

[Meyer 96] J.-A. Meyer. Artificial life and the animat approach to artificial intelligence, pages 325–354. Academic Press, boden, m. edition, 1996.

[Meyer 97] J.A. Meyer. *From natural to artifcial life: Biomimetic mechanisms in animat designs.* Robotics and Autonomous Systems, vol. 22, no. 3, page 21, 1997.

[Minsky 85] M.L. Minsky. The society of mind. Simon & Schuster, 1985.

[Morignot 96] P. Morignot & B. Hayes-Roth. *Motivated Agents.* Rapport technique, Knowledge Systems Laboratory, 1996.

[Morris 78] G.K. Morris, G.E. Kerr & J.H. Fullard. *Phonotactic preferences of female meadow katydids.* Canadian Journal of Zoololgy, 1978.

[Namee 01] B.M. Namee & P. Cunningham. *A Proposal for an Agent Architecture for Proactive Persistent Non Player Characters.* In D. O'Donoghue, editeur, 2th Irish Conference on Artificial Intelligence & Cognitive Science (AICS 2001), pages 221–232, 2001.

[Nareyek 01] A. Nareyek. *Review: Intelligent Agents for Computer Games.* Lecture Notes in Computer Science, vol. 2063, page 414, 2001.

[Nareyek 04] A. Nareyek. *AI in Computer Games.* ACM Queue: Tomorrow's Computing Today, vol. 1, no. 10, pages 58–65, 2004.

[NeverWinter Night 04] The Underdark: NeverWinter Night. http://nwn.bioware.com. Bioware, atari edition, 2004.

[Ortony 88] A. Ortony, G. Clore & A. Collins. The cognitive structure of emotions. Cambridge University Press, 1988.

[Pettre 05] J. Pettre, J.-P. Laumond & D. Thalmann. *A navigation graph for real-time crowd animation on multilayered and uneven terrain.* In The First International Workshop on Crowd Simulation (V-CROWDS '05), 2005.

[Picard 97] R. Picard. Affective computing. The mit press edition, 1997.

[Ponder 03] M. Ponder, T. Molet, G. Papagiannakis, N. Magnenat-Thalmann & D. Thalmann. *VHD++ Development Framework: Towards Extendible, Component Based VR/AR Simulation Engine Featuring Advanced Virtual Character Technologies.* In Comupter Graphics International (CGI) 2003, 2003.

[Pribram 84] K. Pribram. Emotions: A neurobehavioral analysis, pages 13–38. Lawrence Erlbaum Associates, 1984.

[Rank 05] Stephan Rank, Pablo Anjos, Paolo Petta & Ruth Aylett. *What is an Affective Architecture for Situated Agents?* In i.C. (ed.), editeur, Proc. of the Humaine WP7 Workshop: Emotion in Cognition and Action, 2005.

[Rolls 99] E. Rolls. The brain and emotions. Oxford University Press, 1999.

[Rosenblatt 88] J.K. Rosenblatt & D.W. Payton. *A fine-grained alternative to the subsumption architecture for mobile robot control.* In the IEEE/INNS Internationnal Joint Conference on Neuronal Networks, 1988.

[Sannier 99] G. Sannier, S. Balcisoy, N. Magnenat-Thalmann & D. Thalmann. *VHD: A System for Directing Real-Time Virtual Actors.* The Visual Computer, vol. 15, no. 7/8, pages 320–329, 1999.

[Schertenleib 05] S. Schertenleib & D. Thalmann. *Managing High-Level Scripts Execution within Multithreaded Environments.* In Programming Gems Series. Charles River Media, Inc., 2005.

[Seth 98] A.K. Seth. *Evolving action selection and selective attention without actions, attention, or selection.* In Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5, pages 139–146. MIT Press, 1998.

[Sevin 01] E.d. Sevin, M. Kallmann & D. Thalmann. *Towards Real Time Virtual Human Life Simulations.* In CGI '01: Computer Graphics International 2001, pages 31–37. IEEE Computer Society, 2001.

[Sevin 05] E.d. Sevin & D. Thalmann. *An Affective Model of Action Selection for Virtual Humans.* In Proceedings of Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action; symposium at the Artificial Intelligence and Social Behaviors 2005 Conference (AISB'05), 2005.

[Sims 2 04] The Sims 2. http://thesims2.ea.com. Maxix, electronic arts inc. edition, 2004.

[Sloman 87] Sloman. *Motives, mechanisms, and emotions.* Cognition and Emotions, vol. 1, no. 3, pages 217–233, 1987.

[Sloman 97] A. Sloman. *What Sort of Control System Is Able to Have a Personality?* In Creating Personalities for Synthetic Actors, Towards Autonomous Personality Agents, pages 166–208. Springer-Verlag, 1997.

[Sloman 98] A. Sloman & B. Logan. *Cognition and affect (poster): architectures and tools.* In AGENTS '98: Proceedings of the second international conference on Autonomous agents, pages 471–472. ACM Press, 1998.

[Sloman 99] A. Sloman & B. Logan. *Building cognitively rich agents using the SIM_Agent toolkit.* Commun. ACM, vol. 42, no. 3, pages 71–ff., 1999.

[Steegmans 04] E. Steegmans, W. Danny, H. Tom & B. Yolande. *Designing roles for situated agents.* In P. Odell J.a.G. & J.P. Muller, editeurs, The fifth international workshop on agent-oriented software engineering, pages 17–32, 2004.

[Technology 04] W.S. Technology. *http://www.microsoft.com/whdc/system/CEC/HT-Windows.mspx,.* 2004.

[The Elder Scrolls 3 03] Bloodmoon: The Elder Scrolls 3. http://www.elderscro. Bethesda Softworks, ubisoft edition, 2003.

[Thorisson 04] K. Thorisson, C. Pennock, T. List & J. DiPirro. *Artificial intelligence in computer graphics: a constructionist approach.* In ACM SIGGRAPH Computer Graphics, pages 26–30, 2004.

[Tinbergen 50] N. Tinbergen. *The hierarchical organization of mechanisms underlying instinctive behavior.* Experimental Biology, vol. 4, no. 305 - 312, 1950.

[Tinbergen 51] N. Tinbergen. The study of instinct. Oxford University Press, 1951.

[Toates 83] F. Toates. Models of motivations, volume 1. W.H. Freeman & Co., halliday, t. & slate edition, 1983.

[Toates 86] F. Toates. Motivational systems. Cambridge university edition, 1986.

[Tomkins 84] S. Tomkins. Affect theory, pages 163– 195. LaLawrence Erlbaum Associates, 1984.

[Tu 99] X. Tu. *Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behavior.* 1999.

[Tyrrell 92] T. Tyrrell. *Defining the action selection problem.* In the fourteenth annual conf. of the Cognitive Society. Lawrence Erlbaum Associates, 1992.

[Tyrrell 93a] T. Tyrrell. *Computational Mechanisms for Action Selection.* Phd. thesis, University of Edinburgh, 1993.

[Tyrrell 93b] T. Tyrrell. *The Use of Hierarchies for Action Selection.* Adaptive Behavior, vol. 1, pages 387– 420, 1993.

[Tyrrell 94] T. Tyrrell. *An evaluation of Maes' bottom-up mechanism for behavior selection.* Adaptive Behavior, vol. 2, pages 307– 348, 1994.

[Werner 94] G.M. Werner. *Using second order neural connection for motivation of behavioral choices.* In the Third International Conference on Simulation of Adaptative Behavior, pages 154–161, 1994.

[Wilson 87] S.W. Wilson. *Classifier systems and animat problem.* Machine learning, vol. 2, pages 199– 228, 1987.

[Woodcock 00] S. Woodcock. *Game AI: The State of the Industry.* Game Developer Magazine, vol. 7, pages 24–32, 2000.

[Wooldridge 95] M. Wooldridge & N.R. Jennings. *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review, vol. 10, no. 2, pages 115–152, 1995.

[World of Warcraft 05] The World of Warcraft. www.worldofwarcraft.com. Blizzard, vivendi universal games edition, 2005.

Appendix A

Description of the rules used in the main simulation

```
#————————————————————————————————————————
# Essential variables
#————————————————————————————————————————
['food_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['water_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['rest_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['toilet_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['sleep_need', 'essential', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['wash_need', 'essential', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['cook_need', 'essential', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['clean_need', 'essential', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['read_need', 'secondary', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['social_need', 'secondary', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['sport_need', 'secondary', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['waterplant_need', 'secondary', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]

#————————————————————————————————————————
# Motivations
#————————————————————————————————————————
['hunger', 0.0, 0.75, 0.25, 0.15, 0.2]
['thirst', 0.0, 0.75, 0.25, 0.15, 0.2]
['rest', 0.0, 0.75, 0.25, 0.15, 0.2]
```

['toilet', 0.0, 0.75, 0.25, 0.15, 0.2]
['sleeping', 0.0, 0.75, 0.25, 0.15, 0.2]
['washing', 0.0, 0.75, 0.25, 0.15, 0.2]
['cooking', 0.0, 0.75, 0.25, 0.15, 0.2]
['cleaning', 0.0, 0.75, 0.25, 0.15, 0.2]
['reading', 0.0, 0.75, 0.25, 0.15, 0.2]
['communicate', 0.0, 0.75, 0.25, 0.15, 0.2]
['exercise', 0.0, 0.75, 0.25, 0.15, 0.2]
['watering', 0.0, 0.75, 0.25, 0.15, 0.2]

#——————————————————————————
Comportments
#——————————————————————————

["TABLE", 'hunger', 'EAT', [1500, -2, 3500], [["SATISFY", 35]], [], [], 0.75, 0.25, 2.0, 1.0]
["SINK", 'thirst', 'DRINK', [2000, -2, 5000], [["SATISFY", 45]], [], [], 0.75, 0.25, 2.0, 1.0]
["SOFA", 'rest', 'SIT', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["TOILET", 'toilet', 'SATISFY', [5000, -1, 10000], [['WASH', 85]], [], [], 0.75, 0.25, 2.0, 1.0]
["BEDROOM", 'sleeping', 'SLEEP', [-1], [['BALANCE', 30]], [], [], 0.75, 0.25, 2.0, 1.0]
["BATHROOM", 'washing', 'WASH', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["OVEN", 'cooking', 'COOK', [-1], [["EAT", 85], ["EAT1", 45]], [], [], 0.75, 0.25, 2.0, 1.0]
["WORKTOP", 'cleaning', 'CLEAN', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["BOOKSHELF", 'reading', 'READ', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["COMPUTER", 'communicate', 'WRITE EMAIL', [-1], [["PHONE", 40]], [], [], 0.75, 0.25, 2.0, 1.0]
["ROOM", 'exercise', 'DO PUSH-UPS1', [-1], [["REST", 35]], [], [], 0.75, 0.25, 2.0, 1.0]
["PLANT", 'watering', 'WATER', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

#Second comportment at the same location

["TABLE", 'thirst', 'DRINK1', [1500, -1, 2500], [], [], [], 0.75, 0.25, 2.0, 1.0]
["SINK", 'hunger', 'EAT1', [1500, -2, 3500], [], [], [], 0.75, 0.25, 2.0, 1.0]
["SINK", 'cooking', 'COOK1', [-1], [["EAT", 55], ["EAT1", 45]], [], [], 0.75, 0.25, 2.0, 1.0]

["COMPUTER", 'reading', 'READ1', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

["BEDROOM", 'rest', 'SIT1', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

["BATHROOM", 'cleaning', 'CLEAN1', [1000, -1, 2000], [["REST", 25]], [], [], 0.75, 0.25, 2.0, 1.0]

#Second comportment at a different location

["SHELF", 'cleaning', 'CLEAN2', [1000, -1, 2000], [], [], [], 0.75, 0.25, 2.0, 1.0]

["TABLE1", 'cleaning', 'CLEAN3', [1000, -1, 2000], [], [], [], 0.75, 0.25, 2.0, 1.0]

["DESK", 'exercise', 'DO PUSH-UPS2', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

["KITCHEN", 'exercise', 'DO PUSH-UPS3', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

["LIVING ROOM", 'exercise', 'BALANCE', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

["DESK1", 'communicate', 'PHONE', [2000, -2, 5000], [], [], [], 0.75, 0.25, 2.0, 1.0]

["LIVING", 'communicate', 'PHONE2', [2000, -2, 5000], [], [], [], 0.75, 0.25, 2.0, 1.0]

#———————————————————————————————

Actions

#———————————————————————————————

#coorespondance avec les positionskf.posgoal par l'ordre (le numero)

['EAT', -1, 'food_need', 15.0, 0.3, 0.025, 20, 0, "keyeat", 2, ["burger"], 1.0]

['DRINK', -1, 'water_need', 15.0, 0.3, 0.025, 20, 0, "keydrink", 1, [], 0.6]

['SIT', -3, 'rest_need', 25.0, 0.3, 0.025, 20, 0, "", 1, [-1.5, 0.0, -4.0], 1.0]

['SATISFY', -3, 'toilet_need', 15.0, 0.3, 0.025, 20, 0, "", 1, [-1.5, 0.0, 1.8], 1.0]

['SLEEP', -1, 'sleep_need', 100.0, 0.3, 0.025, 20, 0, "keysleep", 1, [], 1.0]

['WASH', -3, 'wash_need', 30.0, 0.3, 0.025, 20, 0, "", 1, ["savon"], 1.0]

['COOK', -1, 'cook_need', 45.0, 0.3, 0.025, 20, 0, "keycook", 10, ["tiroir", "boite", "viande", "fromage", "bun1", "bun2", "spatule", "ketch", "ketchup"], 1.0]

['CLEAN', -1, 'clean_need', 25.0, 0.3, 0.025, 20, 0, "keyclean", 2, ["eponge"], 1.0]

['READ', -1, 'read_need', 30.0, 0.3, 0.025, 20, 0, "keyread", 2, ["book"], 1.0]

['WRITE EMAIL', -1, 'social_need', 20.0, 0.3, 0.025, 20, 0, "keywork", 2, ["chaiseWork"], 1.0]

['DO PUSH-UPS1', -1, 'sport_need', 30.0, 0.3, 0.025, 20, 0, "keypushup5", 1, [], 1.0]

['WATER', -1, 'waterplant_need', 15.0, 0.3, 0.025, 20, 0, "keywater", 2, ["arrosoir"], 1.0]

#Second action at the same location

['DRINK1', -1, 'water_need', 20.0, 0.3, 0.025, 20, 0, "keydrink2", 2, ["gobelet"], 1.0]

['EAT1', -3, 'food_need', 20.0, 0.3, 0.025, 20, 0, "", 1, [10.3, 0.0, -5.4], 1.0]
['COOK1', -3, 'cook_need', 20.0, 0.3, 0.025, 20, 0, "", 1, [10.3, 0.0, -5.4], 1.0]
['READ1', -3, 'read_need', 20.0, 0.3, 0.025, 20, 0, "", 1, [7.6, 0.0, 4.8], 1.0]
['SIT1', -3, 'rest_need', 20.0, 0.3, 0.025, 20, 0, "", 1, [3.4, 0.0, 5.4], 1.0]
['CLEAN1', -3, 'clean_need', 20.0, 0.3, 0.025, 20, 0, "", 1, [-1.1, 0.0, 4.75], 1.0]

#Second action at a different location

['CLEAN2', -1, 'clean_need', 15.0, 0.3, 0.025, 20, 0, "keyclean2", 2, ["eponge2"], 1.0]
['CLEAN3', -1, 'clean_need', 15.0, 0.3, 0.025, 20, 0, "keyclean3", 2, ["patte"], 1.0]
['DO PUSH-UPS2', -1, 'sport_need', 25.0, 0.3, 0.025, 20, 0, "keypushup3", 1, [], 1.0]
['DO PUSH-UPS3', -1, 'sport_need', 30.0, 0.3, 0.025, 20, 0, "keypushup4", 1, [], 1.0]
['BALANCE', -1, 'sport_need', 20.0, 0.3, 0.025, 20, 0, "keyequilibre", 1, [], 1.0]
['PHONE', -1, 'social_need', 15.0, 0.3, 0.025, 20, 0, "keyphone", 2, ["ecouteur"], 1.0]
['PHONE2', -1, 'social_need', 15.0, 0.3, 0.025, 20, 0, "keyphone2", 2, ["portable"], 1.0]

#Compromis behaviors

['EAT AND DRINK', -2, 'TABLE', ['food_need', 'water_need'], 30.0, 0.3, 0.025, 20, 0, "keymeal", 2, ["chair"], 0.5]

['EAT, DRINK AND COOK', -4, 'SINK', ['food_need', 'water_need', 'cook_need'], 40.0, 0.3, 0.025, 20, 0, "", 1, [10.4, 0.0, -5.4], 1.0]

['READ AND PLAY', -4, 'COMPUTER', ['read_need', 'play_need'], 30.0, 0.3, 0.025, 20, 0, "", 1, [7.6, 0.0, 4.8], 1.0]

['SLEEP AND REST', -2, 'BEDROOM', ['sleep_need', 'rest_need'], 60.0, 0.3, 0.025, 20, 0, "keysleep", 1, [], 1.0]

['WASH AND CLEAN', -4, 'BATHROOM', ['wash_need', 'clean_need'], 20.0, 0.3, 0.025, 20, 0, "", 1, [-1.1, 0.0, 4.75], 1.0]

Appendix B

Description of the rules used in the dog simulation

```
#——————————————————————————————
# Essential variables
#——————————————————————————————
['food_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['water_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['rest_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]
['toilet_need', 'basic', 0.25, 2.0, -1.0, 0.0, -1.0, 1.0, -0.75, 0.75, -1.0, 1.0]

#——————————————————————————————
# Motivations
#——————————————————————————————
['hunger', 0.0, 0.75, 0.25, 0.15, 0.2]
['thirst', 0.0, 0.75, 0.25, 0.15, 0.2]
['rest', 0.0, 0.75, 0.25, 0.15, 0.2]
['toilet', 0.0, 0.75, 0.25, 0.15, 0.2]

#——————————————————————————————
# Comportments
#——————————————————————————————
["KITCHEN", 'hunger', 'EAT', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["KITCHEN", 'thirst', 'DRINK', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
```

["CARPET LIVING", 'rest', 'SIT', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]
["DOOR", 'toilet', 'GO OUT', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

#Second comportments at a different location

["CARPET ROOM", 'rest', 'LIE DOWN', [-1], [], [], [], 0.75, 0.25, 2.0, 1.0]

#———————————————————————————
Actions
#———————————————————————————

['EAT', -3, 'food_need', 15.0, 0.3, 0.025, 20, 0, "", 1, [8.4, 0.0, -6.2], 1.0]
['DRINK', -3, 'water_need', 15.0, 0.3, 0.025, 20, 0, "", 1, [8.4, 0.0, -6.2], 1.0]
['SIT', -3, 'rest_need', 25.0, 0.3, 0.025, 20, 0, "", 1, [4.0, 0.0, 3.1], 1.0]
['GO OUT', -3, 'toilet_need', 15.0, 0.3, 0.025, 20, 0, "", 1, [-2.0, 0.0, 0.2], 1.0]

#Second action at a different location

['LIE DOWN', -3, 'rest_need', 25.0, 0.3, 0.025, 20, 0, "", 1, [4.6, 0.0, -6.0], 1.0]

Curriculum Vitae

Name	Etienne de Sevin
Date of birth	18 March 1973 in L'isle Adam, France
Nationality	French
Languages	Good knowledge of English
E-mail	etienne.desevin@epfl.ch, desevin@yahoo.fr

EDUCATION :

1999-2000

Master of Science (Computers graphics)

University of Lyon II (France) and VRLab (EPFL, Switzerland)

Thesis: Making virtual humans motivated

1998-1999

Master of Science (Artificial intelligence)

University of Paris VI (France) and AnimatLab (LIP6, France)

Thesis: Action selection architecture for autonomous animats

1996-1998

Bachelor (Cognitive Sciences)

University of Bordeaux II (France) and AnimatLab (ENS, France)

Thesis: Action selection architecture for motivated animats

PROFESSIONAL ACTIVITIES :

from 2002

Research assistant, VRLab, EPFL, Lausanne

from 2001

System Administrator for windows PCs (VRLab)

april 2000

Implementation of an interactive terminal and a flash web site (Lyon, France)

1999-2000

Teaching basic computer sciences courses (Lyon, France)

PUBLICATIONS :

- E. de Sevin and D. Thalmann, *"An Affective Model of Action Selection for Virtual Humans"*, In Proceedings of Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action symposium at the Artificial Intelligence and Social Behaviors 2005 Conference (AISB'05) , University of Hertfordshire, Hatfield, England, 2005
- E. de Sevin and D. Thalmann, *"A motivational Model of Action Selection for Virtual Humans"*, In Computer Graphics International (CGI), IEEE Computer SocietyPress, New York, 2005
- E. de Sevin and D. Thalmann, *"Let Non-Player Characters Live Their Own Life"*, In Journal of Game Development, 2005 (submitted)
- E. de Sevin and D. Thalmann, *"The complexity of testing a motivational model of action selection for virtual humans"*, In Computer Graphics International (CGI), IEEE Computer SocietyPress, Crete, 2004
- Book contributor: N. Magnenat-Thalmann, D. Thalmann (eds), Handbook of Virtual Humans, John Wiley, 2004
- E. de Sevin, M. Kallmann, and Daniel Thalmann, *"Towards Real Time Virtual Human Life Simulations"*, In Computer Graphics International (CGI), IEEE Computer SocietyPress, Hong-kong, 2001
- M. Kallmann, E. de Sevin, and Daniel Thalmann, *"Contructing Virtual Human Life Simulations"*, AVATARS Workshop, Lausanne, Switzerland, 2000.