A Flexible Multi-Expert Meta-Architecture for Virtual Agents

Etienne de Sevin, Quentin Reynaud, Vincent Corruble

A Flexible Meta-Architecture for Autonomous Virtual Agents

Etienne de Sevin[1] Quentin Reynaud[1, 2] Vincent Corruble[1]
Etienne.de-sevin@jip6.fr Quentin.reynaud@lip6.fr Vincent.corruble@lip6.fr

[1]LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France

[2]Thales Training & Simulation, 1 rue du Général de Gaulle 95520 Osny, France

Abstract :
One of the main motivations in behavioral simulation of virtual agents is the design of agent architectures which are sufficiently flexible, modular, consistent and generic to manage at the same time reactive and cognitive behaviors. We present in this article FlexMex, a flexible multi-expert meta-architecture for virtual agents. The main challenge lies in the structuration and organization of different modules addressing each a specific type of intelligence, each producing its own desires, goals, plans or motivations for behaviors. Our meta-architecture is independent of the module contents. The propagation of the behaviors has to be in a flexible and manageable manner to the decision process. We instantiate it in the context of autonomous pedestrians inhabiting a virtual city according to the real-time, scalability and complexity constraints.

Résumé :
Une des difficultés de la simulation comportementale d'agents virtuels est le design de leur architecture pour qu'elle soit suffisamment flexible, modulaire, cohérente et générique pour gérer en même temps des comportements réactifs et cognitifs. Nous présentons dans cet article, FlexMex, une méta-architecture flexible et multi-expert pour des agents virtuels. Le principal chalenge réside dans la structuration et l'organisation des différents modules qui correspondent chacun à un type spécifique d'intelligence et produisent des comportements à partir de leur propres désirs, buts, plans ou motivations. Notre méta-architecture est indépendante du contenu des modules. La propagation des comportements doit se faire de façon flexible et contrôlable jusqu'au processus de décision. Nous avons instancié notre méta-architecture pour des piétons autonomes peuplant une ville virtuelle en respectant des contraintes de temps-réel, de scalabilité et de complexité.

Keywords: Agent architecture, credibility, virtual agent

1 Introduction

Several trends of research, from cognitive psychology, AI, ethology, to computer games, have contributed over the years techniques that have proved useful to simulate at least some aspects of human behavior. Choosing one over the other can be a matter of adhering to some basic assumptions of the fields, or on some specific functional or non-functional requirements of a given application. Another option, somewhat more pragmatic, is to see how these various contributions from research can be combined in an elegant way in a single framework that draws on each of them depending on the context at hand and is therefore able to simulate a wide variety of behaviors in multiple application domains.

To simulate credible virtual agents, it is now well recognized that an agent architecture has to handle short-term (reactive) behaviors and long-term (cognitive) planning at the same time. Reactive architectures provide quick answers to the environmental pressure (with a low computational cost) and the cognitive ones provide the richness of the behaviors for the virtual agents. Architectures combining both approaches are called hybrid architectures. However, they require also some level of flexibility, modularity, consistency and generality capabilities to produce complex and credible behaviors to simulate autonomous virtual agents. To respect all of these requirements, we claim that a specific structuring and organization of the components in the architecture is needed, and we argue in this paper that this issue can be addressed largely independently from the actual content of the components. That is why we present in this paper what we call a meta-architecture, as the individual components are not described in any detail except for their functionalities, inputs and outputs. Our approach is therefore somewhat related to the notion of control framework as used for example in the TouringMachine (Ferguson, 1992) or CogAff (Sloman, 2001) projects, though our proposal differs significantly.

In this paper, we present FlexMex: a flexible multi-expert meta-architecture for virtual agents fulfilling the requirements mentioned above. This meta-architecture allows to organize the various input components (later named high-level modules) of the architecture, running in parallel and proposing consistent behaviors to a decision module, without any inhibitions and according to their own expertise. These components can be of a reactive nature such as the ones dealing with motivations or emotions, of a cognitive or deliberative nature such as the ones dealing with anticipation or planning, cooperative, etc. depending of the context of the simulation scenario. None of the modules is essential and the meta-architecture is content independent.

After presenting some background on the agent architectures, we focus on key properties that an agent architecture needs and use this analysis grid to evaluate existing ones. We present then our FlexMex meta-architecture in some detail and follow with a description of its application in a collaborative project with an example. Finally, we discuss its advantages and possible limitations and conclude.

2 Background

The interest of hybrid architectures is to combine the strengths of reactive and cognitive approaches. They are widely use in the community. In this section, we base our classification on a Duch paper (Duch, Oentaryo, & Pasquier, 2008).

TouringMachine (Ferguson, 1992) is a three-layer architecture composed of a reactive layer, a planning layer, and a modeling layer. The reactive one directly connects perceptions to actions, it ensure reactivity and rapidness. The planning one generates and executes plans. The modeling one gives reflective and predictive capabilities to the agent by constructing cognitive models of world entities. All these layers have incomplete information, and the actions they propose can be in conflict, that is why a control framework is needed, which has to « behave appropriately in each different world situation ».

The InteRRaP architecture (Müller & Pischel, 1993) separates the decisional process into three steps. The first one is a reactive step: an InteRRaP agent has a set of behaviors, which can respond to its current objective. If none of them matches, the decisional process goes to step two: planning. The agent tries to organize several behaviors in time to reach its goals. If it does not work, the last step is reached: cooperation. The agent tries to contact others agents and asks for help.

The ICARUS architecture (Langley & Choi, 2006) was influenced by SOAR and uses four modules. “Argus” selectively perceives the environment. “Daedalus” plans agent’s behaviors (means-end analysis from GPS (Newell & Simon, 1963)). “Meander” deals with reactive behaviors and executes plans from “Daedalus”. “Labyrinth” stores the agent’s knowledge.

The PECS architecture (Schmidt, 2005) uses four modules too, but they are not organized into a hierarchy. A physical module deals with homeostatic variables, an emotional module is in charge of the agent’s emotional state, a social module manages the cooperation between agents and a cognitive module takes care of the agent's knowledge. They are in permanent competition in order to take control of the agent. The PECS architecture determines which module is the most relevant to deal with the current situation. Afterwards, that module is selected to drive the agent. PECS is a winner-takes-all architecture: only one module drives the agent at any given time.

As we just see in the background, many organizations of high-level modules in hybrid architectures are possible but they are all with some limitations:

- in the InteRRaP architecture, the cognitive modules are used only if the reactive one does not find any solution: the cognitive modules can therefore be bypassed.
- in the ICARUS architecture, the reactive and cognitive modules are organized in a hierarchical manner.

- in the PECS and the TouringMachine architecture, the reactive and cognitive modules are at the same level, but in a winner-takes-all organization.

From our point of view, none of these organizations of components in hybrid architectures is entirely satisfactory. Indeed, they do not meet our four requirements at the same time : flexibility, modularity, consistency and generality. In the next section, we will detail the key proprieties necessary for meeting our four requirements to obtaining credible autonomous virtual agents in comparison with existing hybrid architectures.

3 Key proprieties of our Architecture

Before listing the architecture key proprieties, we want to define some terms in order to avoid any confusion. We will first describe what we call high-level modules, which produce behavior propositions. They include reactive modules which produce short-term behavior proposals based on the agent's motivations or emotions, etc., cognitive modules which propose longer-term behaviors based on anticipation, logical reasoning, learning, etc. They are mainly responsible for the behavior complexity of virtual humans. We considerer them all as high-level modules compared to a decision module which integrates behavior proposals coming from high-level modules together and selects the most appropriate actions.

We will detail our key proprieties to have a generic architecture (see section 3.4) with parallel high-level modules proposing coherent behaviors (see section 3.1) to the decision module, without inhibitions (see section 3.3) and according to their expertise (see section 3.2).

3.1 Module Parallelism

Many hybrid architectures are designed in horizontal layers with a hierarchical organization between high-level modules such as the InteRRaP architecture. Other hybrid architectures authorize multiple communications between components such as the TouringMachine architecture in which information can be injected or removed through the control framework. It means that some modules have to integrate outputs coming from others modules.

Numerical integration is one of the main difficulties in many agent architectures when numerical values are used. These numerical values can be useful to integrate and combine results in order to select the most appropriate behaviors. The values of numerical variables are already difficult to estimate inside modules. Therefore, when some modules have to take into account outputs from other modules, the result can be very complicated to interpret meaningfully. For instance, if the emotional status of a virtual human is 'happy', other modules have to integrate this emotion and combine with their own values to reflect this happiness in their behavior choice. The main problem is to decide how to modify the parameters according to other inputs and how many times to apply them. The complexity of this process is proportional to the number of numerical inputs.

One solution to avoid the numerical integration issue is to place all the high-level modules at the same level and to limit the number of communications between modules. Most of the integration and combination is therefore handled in the decision module. Indeed, independent high-level modules, working in parallel, can control more easily the evolution of their parameters in order to propose more consistent behaviors to the decision module.

3.2 Modularity

Most hybrid architectures contain a predefined and finite list of high-level components, as in TouringMachine, InteRRaP, ICARUS and PECS architectures. It limits the number and the type of high-level modules in these hybrid architectures.

The modularity of the high-level modules can overcome these limitations. Indeed, each module represents one or several capacities of an intelligent agent. For instance, an affective module lets an agent deal with emotions, a cooperation module to collaborate efficiently, a cognitive module to plan complex behaviors and/or anticipate, etc. These high-level modules are experts in their domain and propose behaviors according to their expertise to the decision module. Their number and their type are not a priori limited. In our hybrid architecture, we can adjust the capacities of virtual humans by adding or removing high-level modules according to the role of the agent in the simulation. It defines the complexity and the type of behaviors that the agent can adopt.

Modularity is essential to the diversity, the consistency and the flexibility of the behaviors in high-level modules of hybrid architectures. Their number and their expertise vary depending on the capacities that we need for in the simulation. It can be useful for the scalability of the architecture such as simulating a virtual city inhabited with many pedestrians. With the module parallelism, we obtain modular multi-expert high-level modules working in parallel in the hybrid architecture.

3.3 Free Flow Architecture

Independently of the parallel or hierarchical organization, many hybrid architectures are designed with priorities or competition between high-level modules. They respect a specific order in the control of the components (e.g. reactive before cognitive), such as the InteRRaP architecture. Therefore, cognitive modules are often in practice bypassed. Competition between high-level modules is also often used in hybrid architectures. Only one selected module can control the agent at a given time. They are winner-take-all architectures such as the PECS architecture (see figure 4).

These types of architectures lack flexibility and reactivity. In real-time simulation, the notion of quick adaptation to the changes in the environment is very important to the credibility of the behaviors produced. So the reactive modules should have the possibility to propose adaptive behaviors at any moment in time, even if it requires interrupting the current behavior. A good hybrid architecture should not have to restrict the propagation of the information in order to be reactive and switch rapidly between behaviors. Therefore, the choice between the behaviors of the high-level modules (reactive and cognitive) should not be made before the decision stage. The latter can then consider all the possible behaviors in order to choose the most appropriate one. The notion of Free-flow architectures takes inspiration from free flow hierarchies (Tyrrell, 1993) coming from ethology. It gives more flexibility to the behaviors (Bryson, 2000) and more specifically, allows opportunistic and compromise behaviors. Compromise behaviors are behaviors that are not the best to satisfy any active goal in isolation, but rather offer a good compromise between multiple goals. Free flow architectures are efficient even if there is no hierarchical organization between high-level modules (see section 3.1).

From our point of view, flexibility and reactivity in hybrid architectures are essential. The concept of free flow architecture allows high-level modules to propose behaviors without inhibitions in order to have compromise and opportunistic behaviors. No high-level module can be bypassed or be a priori preferred (as opposed to ICARUS). The choice of the most appropriate behavior is made only in the decision module based on the current context.

3.4 Generality

Most hybrid architectures are designed to work on specific tasks, domains or types of domain even if they can be parameterized to better match a new domain. Hence, and these are only examples, they will either focus on the adequacy with human cognition, on the realism of behavior produced, or on the cost-benefit in terms of amount of computation vs. the credibility of the behaviors in a given context.

In our hybrid architecture, we need a module organization which has to be independent from the module content and the context of the simulation. Indeed, the needed capacities of virtual humans can be instantiated according to the tasks or the domains. None of the capacities is essential. For instance, to simulate some scenarios in a credible virtual city, policemen will need mainly some coordination capacities to patrol in the city and its inhabitants need motivational capacities to be autonomous and affective capacities to react credibly to the city events. All the high-level modules send behaviors to the decision module according to their expertise without any inhibitions (see section 3.3). However, a common formalism has to be followed in order to maintain the flexibility and the diversity of high-level modules (see section 3.2) and to allow their combination. The behavior propositions should always be associated with a priority representing the importance of the behaviors according to the expertise of the high-level modules. It allows the decision module to integrate these behaviors and have the possibility to choose the most appropriate ones.

Hybrid architecture should be designed independently from a context, a task or an application domain and can be instantiated consequently following a specific formalism. We plan to test our architecture in several domains such as video/serious games, security, transport simulation or urban planning (see section 5).

4 Flexible Multi-Expert Meta-Architecture (FlexMex)

Each high-level module produces its own desires, goals, plans or motivations for behaviors or intentions. We define behaviors (or intentions or goals) as high-level tasks such as "organize a train trip", and actions as either intermediate (such as "go to the crossroad") or primitive ("give money to buy ticket"). Behaviors are decomposable in sequence of intermediate and ultimately primitive actions.

Our flexible multi-expert meta-architecture consists of three levels (see figure 1): (1) high-level modules that formulate and propose candidate behaviors, (2) a decision module that arbitrates between candidate behaviors and selects actions, and (3) low-level modules that execute the selected actions.

We summarize the module organization and the functioning of our FlexMex architecture according to the four key priorities. To avoid the limitations of hierarchal organizations of hybrid architectures, we use parallel high-level modules, i.e. they are all at the same level. They can exchange some information if needed but our goal is to limit the number of communications between modules in order to avoid the numerical integration issue (see section 3.1). The high-level modules receive information from the environment. Each one can also access some relevant information such as characteristics of the agent (personality, memory, etc.). Each high-level module is expert in its domain such as affects, logical reasoning, coordination, etc. They have their own algorithm based on homeostasis, resources management, learning, etc. to propose behaviors according to their expertise without any inhibitions, and independently from the other modules. However, none of the high-level modules is in itself critical (FlexMex is operational as long as at least one high-level module is activated). Their number and their type can vary and are not a priori limited.

No selection of behavior is made before the decision module is reached allowing flexibility and reactivity. Opportunistic and compromises behaviors are also possible, as in the free flow hierarchies (Tyrrell, 1993). We use a common formalism for sending the behavior propositions in order to have the possibility to add or remove easily high-level modules. The high-level modules output candidate behaviors with an associated priority. The latter represents how important it is, from the point of view of the expertise of the originating module, that this behavior be selected. Let us note that each module can output several (behavior, priority) couples simultaneously. These priorities are used in the decision module for integrating the propositions of behaviors and for choosing the most appropriate actions.

A key originality of our approach is to use modular, parallel, free flow and generic organization of high-level modules in a hybrid meta-architecture. Therefore, we need to have an integration phase in the decision module, which is somewhat challenging. It has to

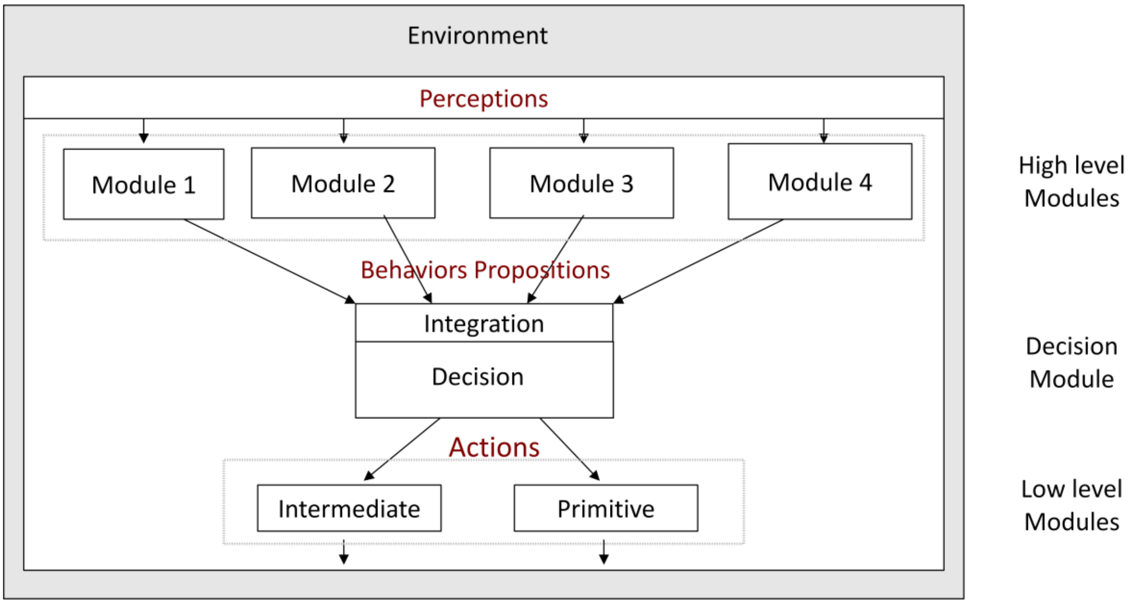

FIG. 1. Flexible multi-expert meta-architecture for virtual agents.

compare and combine all the candidate behaviors after adapting their priorities based on the context of the agent. The difficulty lies in the heterogeneity of the behaviors coming from different modules. However, we centralize the complexity in the decision module instead of having it in each high-level module. We use a generic integration function in order to combine all the behavior propositions independently from their origin. One solution is to integrate behaviors during the decomposition of the behaviors into sequences of actions. The descriptions of the integration function and of the decision module are out of the scope of this article. Please refer to (Reynaud, de Sevin, Donnart, & Corruble, 2012) and forthcoming publications for more details.

5. An pedestrian hybrid architecture

In this section, we instantiate our flexible multi-expert meta-architecture for virtual agents with the agent architecture used in the collaborative Terra Dynamica project. This project aims at building an artificial intelligence framework for the simulation of human-like agents in virtual urban environments to populate virtual cities with credible and autonomous pedestrians.

Terra Dynamica is faced with a number of significant challenges:

- generality: the possibility to use the same hybrid architecture in several domains such as video games, security, transports and urbanism simulations.
- scalability: a great number of agents might be required in the simulation.
- real-time: agent's response time could be critical.
- rich environments: large cities are complex systems, because of their dynamicity and wide range of interactions.

To instantiate our meta-architecture, we have to determine which capabilities are needed by the virtual pedestrians to populate a virtual city in a manner. These capacities can be grouped in five high-level modules (see figure 2):

- a motivational module proposes behaviors in reaction to the evolution of internal variables such as the nutritional level. It represents the reactive, present-oriented intelligence of the agent and is essential for the autonomy of the virtual pedestrians. The dynamic of the motivations' homeostasis is managed to urge the pedestrians to act to satisfy the motivations and provide them with a real autonomy (De Sevin & Thalmann, 2005).

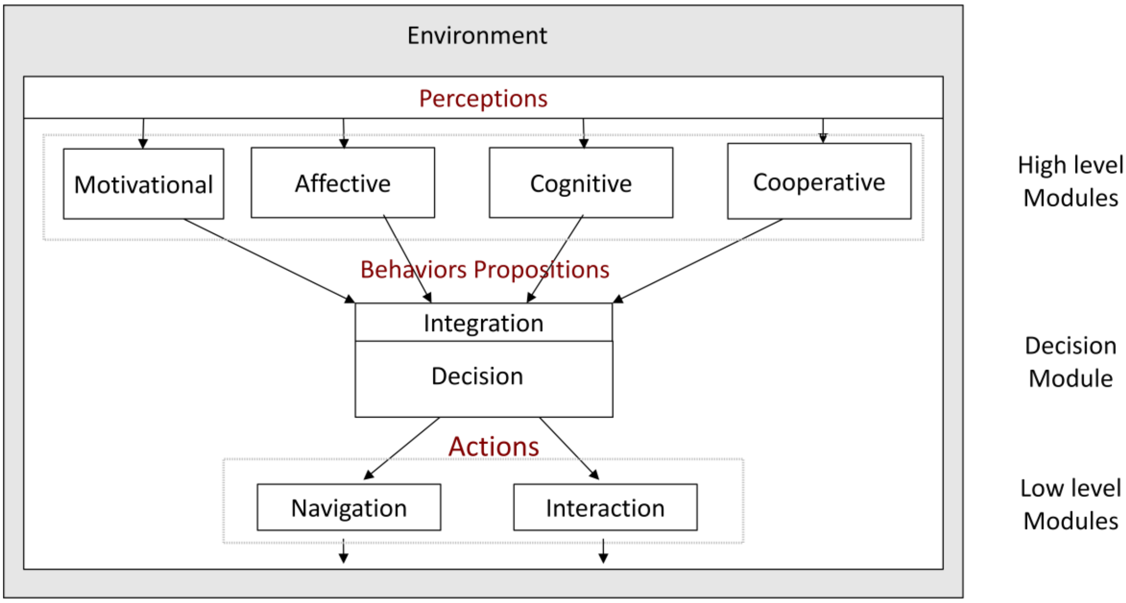

FIG. 2. A pedestrian architecture based on our meta-architecture

- an <u>affective module</u> proposes behaviors in reaction to external events in a subjective manner. It represents also the reactive, present-oriented intelligence of the agent and is essential for the credibility of the virtual pedestrians. This module lets an agent react subjectively and emotionally to some simulation events, for example a fire or a riot. We use a model based on a theory of conservation and acquisition of affective and material resources (Campano, de Sevin, Corruble, & Sabouret, 2011). It can also enhance the social interactions and the adaptation of the virtual pedestrians.
- a <u>cognitive module</u> elaborates plans to reach specific complex goals. It represents the deliberative, future-oriented intelligence of the agent and can be allocated computational resources depending on the current time pressure (Reynaud, de Sevin, Donnart, & Corruble, 2012):
    - anticipation: predicting the next choices of behaviors of the virtual pedestrians and proposes alternative behaviors to the decision module which can be more appropriate in a long-term perspective.
    - long-term planning: optimizing the behaviors coming f rom the reactive modules over the time by learning. It also designs complex course of action to achieve complex goals.
- a <u>cooperative module</u> deals with collective goals. It corresponds to the collective intelligence of the virtual pedestrians (Poulet, Corruble, Seghrouchni, & Ramalho, 2011):
    - coordination: synchronizing virtual pedestrians on shared goals and problems such as police patrol.
    - collaboration: virtual pedestrians can work together to achieve shared goals or tackle problems that they cannot solve alone.

The low-level modules deal with intermediate actions such as navigation ("go to location x") and primitive actions such as interactions with the environment ("buy y").

We can vary the number and the type of agent's high-level modules for the scalability of the simulation depending of their roles. We have instantiated our meta-architecture for an urban simulation. This process can be done with other domains or problems.

6. Conclusion and Perspectives

In this article, we presented FlexMex: a flexible multi-expert meta-architecture for virtual agents meeting some important flexibility, modularity, consistency and generality requirements. These requirements are essential for obtaining credible behaviors for autonomous virtual agents in terms of complexity, adaptability, diversity and reusability. The meta-architecture is composed of high-level modules, running in parallel and proposing coherent behaviors to the decision module, without any inhibitions and according to their expertise.

While individual high level components of our architectures have already been implemented and evaluated separately, we have to finalize their integration in a single instantiated FlexMex architecture in our collaborative project to evaluate it fully. Then, we plan to evaluate the implication and the importance of our four key properties: the module parallelism, the modularity, the free flow organization and the generality. The architecture is to be used in several applications in the video game, security, transport and urban planning domains in the Terra Dynamica project. We also wish to compare FlexMex in more details with well-known architectures such as the PECS, InteRRaP and ICARUS architectures. We are currently working on a generic behavior integration in the decision module for our FlexMex meta-architecture.

Acknowledgements

This research is funded by the Ile de France Region within the Terra Dynamica Project (FUI8) supported by CapDigital and Advancity business clusters.

Reference

Campano, S., de Sevin, E., Corruble, V., & Sabouret, N. (2011). Simulating affective behaviours: an approach based on the COR theory. *Affective Computing and Intelligent Interaction*. Memphis, USA.

de Sevin, E., & Thalmann, D. (2005). A motivational model of action selection for virtual humans. *Computer Graphics International* (Vol. 2005, pp. 213–220).

Duch, W., Oentaryo, R. J., & Pasquier, M. (2008). Cognitive Architectures: Where do we go from here? In P. Wang, B. Goertzel, & S. Franklin (Eds.), *Proceeding of the 2008 conference on Artificial General Intelligence 2008.* (Vol. 171, pp. 122–136). IOS Press.

Ferguson, I. A. Touring Machines: autonomous agents with attitudes. , 25 Computer pp. 51–55 (1992). doi:10.1109/2.144395

Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. In A. Cohn (Ed.), *Proceedings Of The National Conference On Artificial Intelligence* (Vol. 21, p. 1469). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Müller, J. P., & Pischel, M. (1993). The agent architecture inteRRaP: Concept and application. (RR-93-26), 99.

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and Thought* (pp. 279–293). McGraw-Hill.

Poulet, C., Corruble, V., Seghrouchni, A. E. F., & Ramalho, G. The Open System Setting in Timed Multiagent Patrolling. , 2 2011 IEEEWICACM International Conferences on Web Intelligence and Intelligent Agent Technology 373–376 (2011).

Reynaud, Q., de Sevin, E., Donnart, J. Y., & Corruble, V. (2012). A cognitive module in a decision-making architecture for agents in urban simulations. *Workshop on Cognitive Agents in Virtual Environments (CAVE). AAMAS*. Valencia, Spain.

Schmidt, B. (2005). Human Factors in Complex Systems The Modelling of Human Behaviour. *Nature*, *4*(Cd), 5–14.

Sloman, A. (2001). Varieties of affect and the cogaff architecture schema. *Proceedings Symposium on Emotion Cognition and Affective Computing AISB* (Vol. 1, pp. 39–48).