
FlexMex: Flexible Multi-Expert Meta-Architecture for Virtual Agents

Etienne de Sevin¹
Quentin Reynaud^{1, 2}
Vincent Corruble¹

ETIENNE.DE-SEVIN@LIP6.FR
QUENTIN.REYNAUD@LIP6.FR
VINCENT.CORRUBLE@LIP6.FR

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

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

Abstract

One of the key issues in the field of virtual agents is the design of agent architectures which fulfill the conditions necessary to manage at the same time real-time reactive behaviors and longer term cognitive abilities. 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 several modules addressing each a specific type of intelligence, each producing its own desires, goals, plans or motivations for behavior. The meta-architecture presented here is at a level which is largely independent of the module contents. The propagation of the behavior proposal through the architecture down to the level of a final decision step has to be in a flexible and manageable manner. We instantiate our proposal in the context of a projecting aiming at animating autonomous actors living in a virtual city, while respecting constraints of credibility, real-time, scalability and reuse of the architecture.

1. Introduction

Several trends of research in cognitive psychology, AI, ethology and computer games have provided over the years several models and techniques which have been 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 abilities such as planning at the same time. Reactive architectures provide quick answers to the environmental pressure (with a low computational cost) and the cognitive ones allow for the richness of behaviors for credible virtual agents. Architectures combining both approaches are called hybrid architectures.

This work is being carried out within the context of TerraDynamica, a collaborative project aiming at building an artificial intelligence framework for the simulation of human-like agents in virtual urban environments.

Terra Dynamica¹ is faced with a number of significant challenges:

- Credible virtual pedestrians: generation of their own goal, subjective reaction to city events, anticipation and planning of their behaviors, etc.
- Scalability: a great number of agents might be required in the simulation. Thousand of pedestrians should be simulated at the same time.
- Real-time: agent's response time to the city events could be critical for the credibility of the agent behaviors.
- Reuse of a generic framework: the possibility to use a single architecture and platform in several application domains such as video games, security, transports and urban simulations, each with its own goals and requirements.

We inferred from an initial analysis that our hybrid meta-architecture FlexMex should have four key properties to face these challenges and produce reactive and complex behaviors to simulate credible autonomous virtual agents :

- *Flexibility*: distributed management of reactive and complex agent behaviors. The meta-architecture should be able to blend them to obtain credible behaviors.
- *Modularity*: no limit on the number and types of modules proposing behaviors, which depend on the agent complexity in the simulation.
- *Consistency*: the modules proposing behaviors are independent and running in parallel. They all work as specialized expert to propose behaviors that seem appropriate for their own point of view.
- *Generality*: allowing the reuse of the architecture. The meta-architecture should be independent of the context of the application and needs a specific instantiation.

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 here 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. This architectural design fulfill the requirements mentioned above and 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. and can be activated or deactivated depending on the context of the simulation scenario. None of these modules is therefore essential. The meta-architecture is content independent and need to be instantiated in order to be implemented for specific applications.

¹ <http://www.terrodynamica.com>

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

2. Background

Two main approaches coexist concerning decision-making architectures: the reactive approach (Brooks, 1986) and the cognitive one (Langley, Laird, & Rogers, 2009). A reactive agent acts in response to internal or external stimuli. An internal representation can be used but the agent handles no reasoning. A cognitive agent behaves by reasoning on a symbolic representation of itself and its environment. More recently, the concept of hybrid architecture has emerged. Its principle is to combine the two latter approaches and add up their advantages. In this section we base our classification of architectures on Duch, Oentaryo, & Pasquier (2008). As the number of agent architectures is important, we limit our classification to architectures for virtual agent.

2.1 Reactive Architectures

Initially, reactive architectures were developed to model simple behaviors. Reactive architectures are often strongly related to the bottom-up trend, which advocates the thesis that intelligence can spring from cooperation between simple modules. A good example of this trend is Brooks' subsumption architecture (Brooks, 1986) or the Animat approach (Meyer, 1996). In Brook's hierarchical architecture, several modules (each in charge of a specific behavior) judge the suitability of their own activation. In order to avoid conflicts, the modules are strictly prioritized: a high level module inhibits all lower level modules.

Maes proposed a system where each behavior decides on its own activation using an activation level (Maes, 1990). These levels change dynamically and receive bonuses or penalties which are favoring multi-goals, opportunistic behaviors, while avoiding conflicts. DAMN (Rosenblatt, 1997) is also an important work based on a voting mechanism. A DAMN agent has one module for each possible behavior (follow a road, avoid obstacles, maintain internal variable...). Each module grades each feasible action, deemed relevant to its interest. The top-rated action, the most relevant for all behaviors, is selected.

Based on Rosenblatt and Payton's work (Rosenblatt & Payton, 1989), Tyrrell proposed another way to handle decision or action selection (Tyrrell, 1993) by decomposing behaviors into sub-behaviors until "elementary actions" are dealt with. In this "free-flow hierarchy" approach, at each step, all relevant stimuli are taken into account, and a key idea is that an agent does not take any decision before the final elementary action level is evaluated. The final decision is delayed in order to allow the agent to make compromises (actions which are useful to more than one behavior/goal).

2.2 Cognitive Architectures

The other decision-making modeling approach is the cognitive trend. Two key cognitive architectures are SOAR (Laird, Newell, & Rosenbloom, 1987) and ACT-R (Anderson et al., 2004). They are based on problem solving and the use of production rules. Behaviors are the

result of planning functions that use different types of memory. For example, SOAR creates the agent working memory (which is the mind configuration used to solve the agent's current problem) based on three types of knowledge: procedural memory which contains production rules; semantic memory which stores facts; and episodic memory which saves old working memories (in order to reuse it in future similar situations).

PRODIGY (Veloso et al., 1995) is an architecture that integrates planning with multiple learning mechanisms. The knowledge is stored in a symbolic way in first-order predicate logic. This architecture uses a planning approach inspired from STRIPS (Means-Ends analysis).

The BDI approach (Rao & Georgeff, 1995) is currently widely used. A BDI agent is made up of three components. It consists of desires (D) that can be conflicting or can be irrelevant in the current situation. It has beliefs (B) about itself and its environment and uses them to select and then work towards its intentions (I), which are helping the agent to accomplish its desires.

2.3 Hybrid Architectures

Hybrid architectures try to combine the strengths of reactive and cognitive approaches. For example, TouringMachine (Ferguson, 1992) is a three-layer architecture composed of a reactive layer, a planning layer, and a modeling layer (see Figure 1). The reactive one directly connects perceptions to actions, it ensure reactiveness and swiftness. 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 conflicting, that is why a control framework is needed, which has to « behave appropriately in each different world situation ».

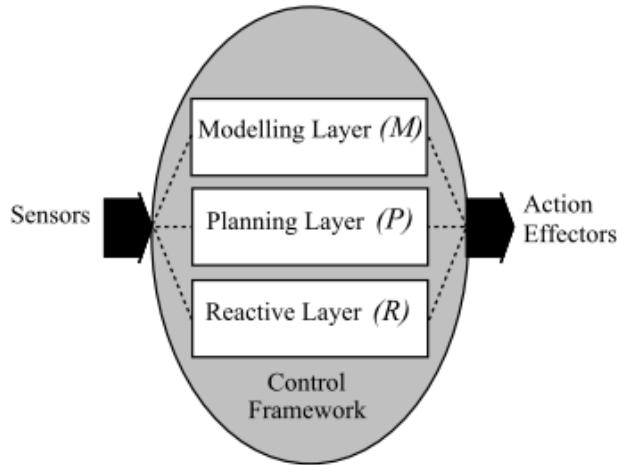


Figure 1. TouringMachine architecture.

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.

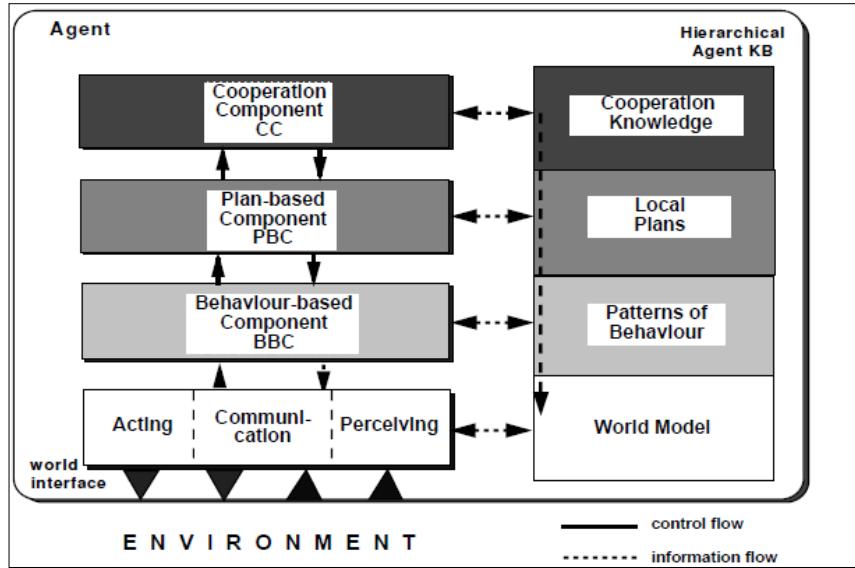


Figure 2. InteRRaP architecture.

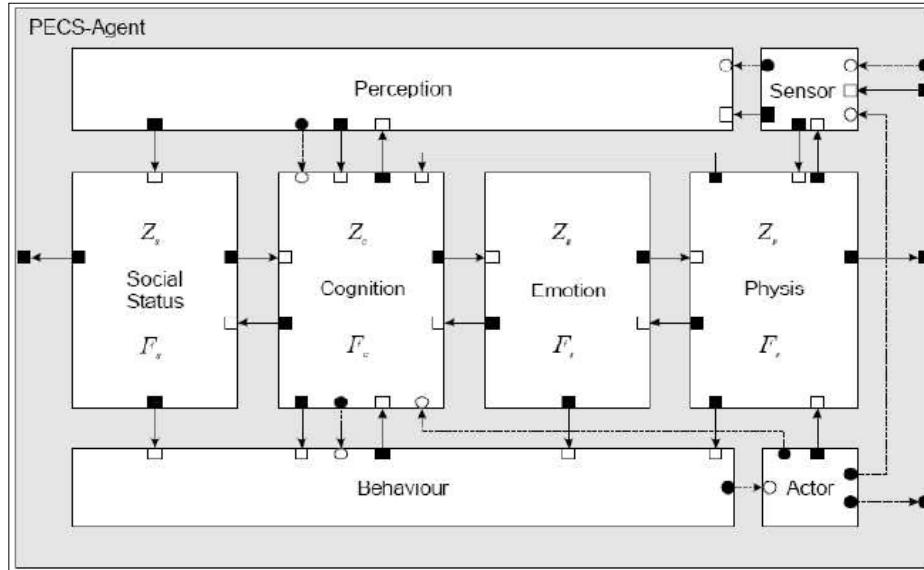


Figure 3. PECS architecture.

The ICARUS architecture (Langley & Choi, 2006) uses four modules (see Figure 2). “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 (see Figure 3). 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 have seen in the related work section, 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.
- They all use only predefined modules and they do not use a distinct decision module.

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 target proprieties, we want to define some terms in order to avoid any confusion. We will first describe what we call *high-level modules*, which output behavior proposals. 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 consider 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.

In our meta-architecture, we made the choice of combining reactive and cognitive abilities. Indeed our agents need to handle quick adaptations to changes in the environment and have to produce credible behaviors for the virtual humans implying behavior complexity and planning in cognitive modules. This planning has to continue until its end but can also be interrupted if needed. That places us in the hybrid approach.

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 Flexibility

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 3).

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. 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.

3.2 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 with other agents, 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. With FlexMex, one can adjust the capacities of agents in a simulation by activating or deactivating high-level modules according to the role of the agent in the simulation and computational resources at hand. It impacts 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 in the simulation. It can be useful for the scalability of the architecture such as simulating a virtual city populated with large numbers of inhabitants.

3.3 Consistency

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 (see Figure 1) in which information can be injected or removed through the control framework. It means that some modules have to integrate numerical outputs coming from others modules.

Numerical integration is one of the main difficulties in many agent architectures when numerical values are used to communicate between modules. 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 increases quickly with 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 a single 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.

3.4 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 often 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, its inhabitants need motivational capacities to be autonomous and affective capacities to react credibly to the city events. All the high-level modules propose behaviors to the decision module according to their expertise without any inhibitions (see section 3.3). However, a common formalism and representation has to be respected in order to maintain the consistency and the diversity of high-level modules 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 ideally be designed independently from a specific task or application domain, and can be instantiated accordingly. We plan to test our architecture in several domains such as video/serious games, security, transport simulation or urban planning.

4. Flexible Multi-Expert Meta-Architecture (FlexMex)

4.1 Principle

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 4): (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 target properties. To avoid the limitations of hierachal 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.

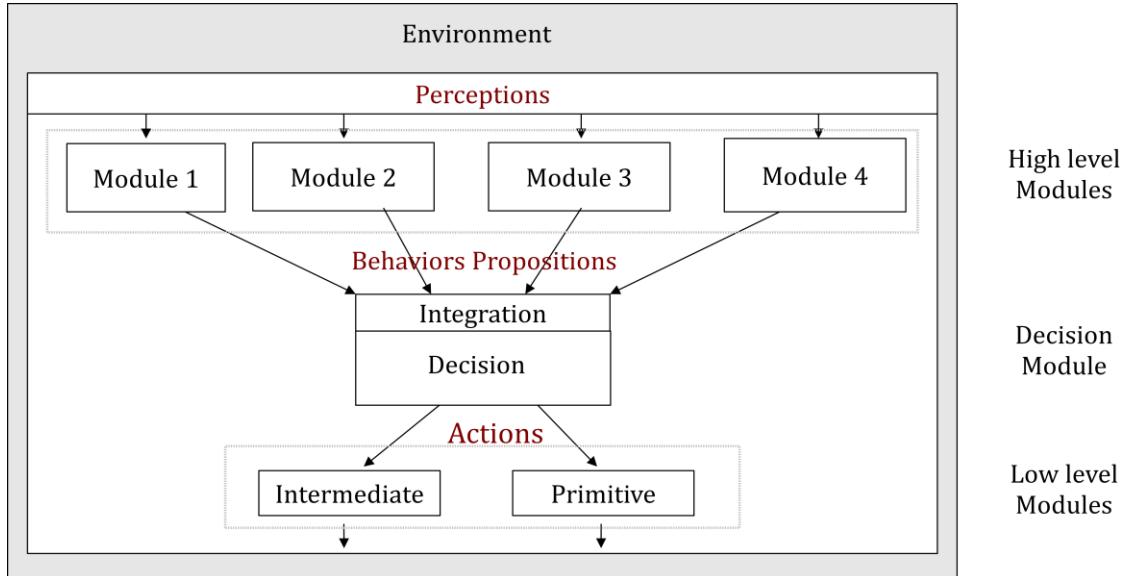


Figure 4. Flexible multi-expert meta-architecture for virtual agents.

No final decision is made before the decision module is reached allowing flexibility and reactivity. Opportunistic and compromises behaviors are therefore possible, as in the free flow hierarchies (Tyrrell, 1993). 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 best actions.

4.2 Decision Module

Basically our decisional module is based on free-flow hierarchy by (Tyrrell, 1993) which is itself based on (Rosenblatt & Payton, 1989). The decisional module takes as input prioritized behavior proposals and gives as output a small set of elementary actions which should be immediately and simultaneously executed by the agent. Each behavior proposal is decomposed into a sequence of elementary actions then emerged compromise behaviors, and finally the preferred action is chosen. However we introduced some major changes.

First of all, we add an integration phase at the beginning of the decisional process. In fact, our architecture combines the decisional module with a modular set of high level modules. One result is that the behavioral proposals can be very different from each other (for example a cognitive module could propose a mission, resulting from many actions during a long time frame; and a “reflex” module could propose in the same time an elementary action such as “blink”). This particularity is not an issue because of the free-flow decisional process feature: no final decision is made before all behavior proposals are decomposed into elementary actions. So each proposal is integrated in the same way. The real problem in this phase is the potential difference of prioritizing scales. High level modules could be seen as black box, and each has its own rules to calculate priorities. Each behavior proposal priority is between 0 (considered as the absence of priority) and 1 (an absolute priority, overriding all others). If there is disequilibrium between high-level modules, one can modify the incriminated modules, or take it into account when the weights are created. Furthermore, automatically learn how to deal with disequilibrium is an excellent path in order to add some learning in our architecture. One way to do so is to use the anticipatory planning, for example by adjusting the module’s weights each time the anticipatory planning detects a preferred behavior which was proposed by the module B instead of the current behavior which was proposed by the module A.

One other major difference is that we do not only compare elementary actions between each others to find the more accurate: we compare entire alternatives. That is to say that instead of a classical decomposition into elementary actions, during the decomposition we search all the possible paths that can be chosen to accomplish the agent’s goals. Indeed, if it can be relevant to make an animal or a robot thinks in short term range only (a couple of actions), it is impossible in order to guarantee credibility to a human behavior. It seems obvious that humans are able to choose their behavior not only by comparing them step after step. That is why our decisional process compares behaviors in their totality. This phase is very important because, according to its knowledge, an agent may have various possibilities to fulfill each goal, and this is at this point of the decisional process that we compare the most relevant alternatives. The comparison takes into account the duration, the cost, the length of the journey, and of course the preferences of the agent. Compromises are directly made between alternatives though the elementary actions they have in common.

5. Application in a Collaborative Project

5.1 A Hybrid Architecture for Credible and Autonomous Pedestrians

In this section, we instantiate our flexible multi-expert meta-architecture for virtual agents with the agent architecture used in the collaborative Terra Dynamica project². 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:

- Credible agents: they should generate their own goal, react subjectively to city events, anticipate and plan of their behaviors, etc. When users look at the city simulation, pedestrians should have credible behaviors at each moment in time and over the time. It implies that the pedestrians have to react quickly to all the events and also able to do complex behaviors. They should give the illusion of "living their own lives" such as eating in a restaurant with friends at lunchtime, doing shopping on the way to go home or greeting others pedestrians on the way. Hybrid architecture need flexibility in their behavior selection to give the credibility to virtual pedestrians.
- Scalability: a great number of agents might be required in the simulation of a virtual city. Only virtual pedestrians where the users look at are showed. As the users can change their point of view, other pedestrians should be simulated even if they are not directly visible. However simulate to many pedestrians at the same time is not possible actually. The hybrid architecture should be enough modular to manage the complexity of the pedestrians according their roles in the simulation and where the users look at.
- Real-time: agent's response time could be critical. To be credible, the pedestrians should react in real-time to internal and external events. For example, if there is an explosion, pedestrians should be afraid and flee to save their lives. If not, they will be not credible. It is similar with the opportunism. The pedestrians should consider all information to satisfy their goals even they have to change their current plans. Managing real-time pedestrians in an large virtual city requires a consistent and efficient hybrid architecture.
- Scope: the possibility to use the same architecture in several domains such as video games, security, transports and urban simulations. Specific scenarios are defined by industrial partners to evaluate the model to be used in several applications such as urban planning, management of city traffic, pursuit in a town or demonstration simulation. The high-level modules are chosen according the scenario and the role of the pedestrians.

Our FlexMex architecture is well positioned to face these challenges. The reuse of our hybrid architecture is facilitated because it is not designed for a specific task or domain and can be instantiated to match new simulation problems. With the modularity of our meta-architecture, we can manage the scalability challenge. We can define (possibly for each agent) the number and the type of high-level modules in the architecture according to the role of the pedestrians and to the

² <http://www.terrodynamica.com>

focus of the users in the simulation. As our architecture has a hybrid, parallel and free flow organization of the high-level modules, it can handle real-time response time. The complexity of the rich environment and corresponding behaviors are also handled with the cognitive part of the hybrid architecture.

To instantiate the high-level modules of our meta-architecture, we have to determine which capabilities are needed by the virtual pedestrians to populate a virtual city in a manner. The instantiated architecture has to produce behaviors satisfying some properties:

- Adaptability: they have to react quickly to the simulation dynamics in real-time both internally (e.g. motivations), and externally (environment, city events).
- Flexibility: ongoing behaviors should be interrupted when necessary and compromise behaviors should be preferred in the choice process.
- Complexity: they can have complex behaviors resulting from planning in order to be credible.
- Anticipation: they can predict their future behaviors to optimize the choice of the more appropriate behaviors.
- Autonomy: they can generate their own goals in order to give the impression of living their own lives.
- Consistency: the behavior generated by the architecture have to be consistent at each moment in time and also over time.
- Collaboration: they can interact, cooperate and collaborate with other virtual pedestrians to deal with collective problems.

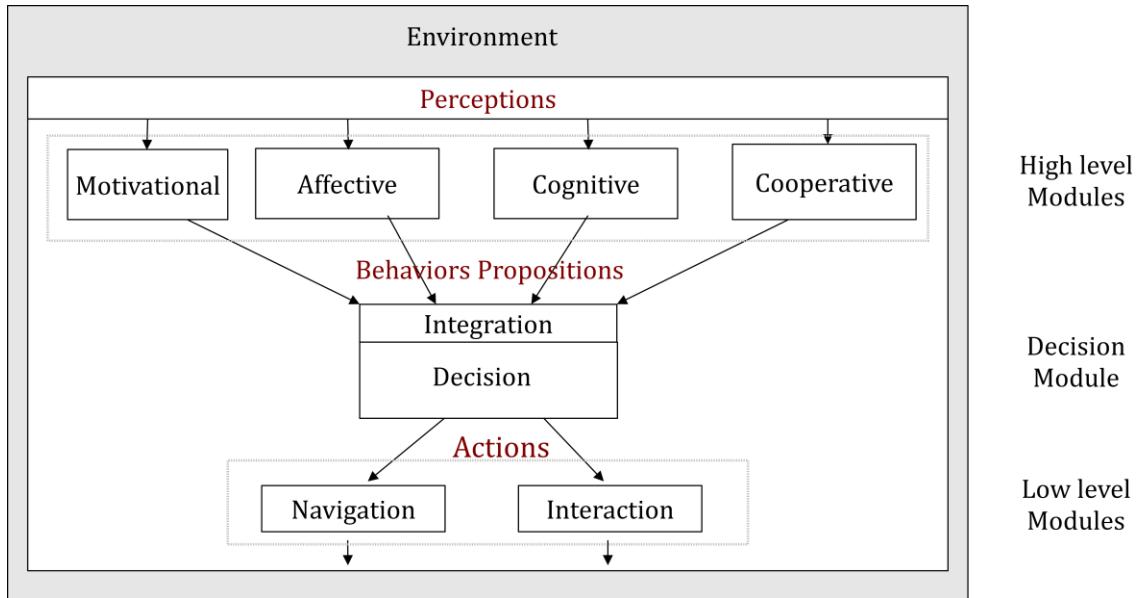


Figure 5. A pedestrian architecture based on our meta-architecture

The capacities can be grouped in four high-level modules in our instantiated architecture for credible virtual pedestrians (see Figure 5):

- A motivational module proposes behaviors in reaction to the evolution of internal variables such as the energy 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 their current motivations (de Sevin & Thalmann, 2005).
- An affective module 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 cognitive module 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 actors and proposes alternative behaviors to the decision module which can be more appropriate in a long-term perspective.
 - Long-term planning: designs complex course of action to achieve complex goals. Also improves on the behaviors proposed by the reactive modules based on past experience.
- A cooperative module deals with collective goals. The virtual actors can work together to achieve shared goals or tackle problems that they cannot solve alone. A good example of this is the coordination problem for collective tasks such as multi-agent patrolling (Poulet, Corruble, Seghrouchni, & Ramalho, 2011), which can be relevant to simulate police patrols or post-disaster rescue operations in cities.

The parallel multi-expert high-level modules receive as inputs the information of the environment to update their world state. They compute their behavior proposal(s) with their own algorithms (based on homeostasis, affective models, planning, etc.). They are all expert in their behavioral domain and represent specific capabilities of the virtual actors. The more high-level modules an agent has, the more it can adapt to the environment, adopt complex behaviors or interact with other pedestrians in order to be more credible. However, not all the agents have to be very smart so we can handle more agents if they have fewer high-level modules active..

The high-level modules propose to the decision module behaviors which they consider important at the moment in time without any inhibitions. It lets them react quickly to the simulation changes, manage the real-time time responses and interrupt the current behavior if necessary. The high-level modules do not know if their behavior proposals will be selected by the decision module. The behavior priorities are propagated in the architecture to help the integration

algorithm of the decision module to combine behaviors and choose the most appropriate primitive actions taking into account the context of the virtual agent such as time, distance, etc. 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 have instantiated our FlexMex meta-architecture for an urban simulation by defining capacities and the high-level modules associated in order to populate a virtual city with credible and autonomous pedestrians. This process can be done with other domains or problems.

5.2 Scenario Examples

5.2.1 Security: Demonstration in a Virtual Town

The goal of this scenario example is to simulate protesters walking on a road that is monitored by the police. Some rioters are at the end of the demonstration and break some shop windows. The scenario can be used to help the police manage demonstrations and train the police by simulating different situations of demonstrations. All these virtual agents have to behave in a credible fashion and therefore need specific capabilities:

- The protesters:
 - An affective module evaluates whether they wish to join the demonstration, to adapt to the events and to decide whether they join the rioters, continue the demonstration or leave it.
 - A cognitive module anticipates on the evolution of the demonstration and on the actions of the rioters/policemen.
 - A cooperative module is used to collaborate and remain coordinated with other protesters.
 - A motivational module is used to provide basic autonomy, urge to manifest and survival.
- The policemen:
 - A cooperative module to remain coordinated and to patrol around the demonstration.
 - A motivational module provides autonomy and reactivity to the rioters' acts of aggression.
 - A cognitive module to plan arrests of some rioters and other complex actions.
- The rioters:
 - A motivational module to be autonomous, break the showcases and escape from the policemen.

We can create some profiles of agents according to the capabilities that they need in the simulation scenario. The important virtual humans (the protesters) have several high-level modules in parallel while the secondary virtual humans (the rioters) are mostly reactive. For the latter, only behaviors coming from the motivational module will be taken into account. For the protesters, the decision module has to integrate all the possible behaviors coming from several high-level modules. We can create a credible simulation scenario of a demonstration in a flexible and easy way and focus on scenario priority in term of virtual human capacities. These profiles can be useful for the scalability of the application.

5.2.2 Video Game: Terrorist Attack

The goal of this application is to catch a dangerous terrorist who wants to plant a bomb in a crowded place in the virtual city for obscure motives. The player has to help the policemen to catch the terrorist. He/she has access to all the information available to the police and can observe the crowded place. The police patrols in the virtual city to try to avoid the terrorist attack and can question pedestrians if they have some doubts.

Several type of autonomous agents have to behave in a credible fashion and therefore need specific capabilities:

- The terrorist:
 - A cognitive module to plan how to plant the bomb, modify their plans if something appends, anticipate the possible problems and remain unnoticed by the police.
 - An affective module to be able to act credibly in presence of stress, anger and fear.
 - A motivational module to be autonomous and survive.
- The policemen:
 - A cognitive module to anticipate what the terrorist is doing and to plan how it can plant its bomb
 - A cooperative module to collaborate with the users and other policemen in order to catch the terrorist.
 - A motivational module to be autonomous and credible.
- The pedestrians:
 - An affective module to interact with policemen during a questioning and express fear if they see the bomb.
 - A motivational module to be autonomous and give the illusion that the pedestrians "live their lives".

The users monitor all the actions of the autonomous policemen and can send some information to them to help catch the terrorist such as "the criminal could be this virtual agent with the briefcase". The policemen will question this agent in order to know if it is the terrorist or not. However the terrorist is cunning and will not be caught easily.

This scenario could be also used in the context of security analysis to simulate a terrorist attacks and help train the police in this type of situations in virtual city.

6. Discussion

6.1 Advantages

The main advantages of our flexible multi-expert meta-architecture are the flexibility, consistency, generality and modularity. We limit the dependencies between high-level modules and the numerical integration issue. All the behaviors are proposed without any inhibition by the high-level modules. It allows opportunism, compromise behaviors and the possibility to interrupt the current behavior if another is more urgent. If the situation changes rapidly, as we have several behaviors active in parallel, the meta-architecture can focus on another behavior that is more

appropriate with respect to the new situation. It enhances the adaptation of our agents to their environment.

In addition to the adaptation, our agents can adopt complex behaviors as a result of their cognitive module. If they can only adapt to the environment changes, they behave in a simple manner and the users can detect it. Our meta-architecture can manage simple adaptive behaviors in parallel with smarter behaviors such as anticipation or long-term behavior planning. It gives consistency to the virtual human over time instead of reactive behaviors at each moment in time. Moreover, each high-level module is expert in its domain and it enhances the coherence of the behaviors proposed to the decision module.

None of the high-level modules is essential. Each one gives a form of intelligence to our agents: short-term adaptation with reactive modules, long-term adaptation with cognitive ones and social adaptation with cooperative ones. We can use them depending on our purpose, on the targeted degree of the complexity and credibility of our agents, and on the available computational resources. For instance, important characters in a simulation have all high-level modules activated and can do complex behaviors while secondary characters have only motivational module to be autonomous. It is then possible to determine profiles of our agents according to their roles in the simulation so as to configure them consequently.

6.2 Limitations

In FlexMex, the decision module can be challenging. Indeed it has to integrate several behaviors and the associated priorities coming from heterogeneous high-level modules. However the main advantage is that we limit the complexity inside the high-level modules which can be more difficult to deal with. The second advantage is that we can monitor the complexity in the more efficient and flexible way in the decision module. We choose to combine behaviors during their decomposition in atomic actions using free flow hierarchies allowing compromise solutions (see section 4.2). The decision module is currently under final implementation and the whole platform is about to be evaluated with several application scenarios.

The second limit relates to the connections between modules. They have to be limited to avoid the numerical integration issue (see section 3.3). Our solution is that the modules can know only the inputs and the outputs of other modules. Variables in modules can be modified only by the modules them-selves. Moreover the inputs and the outputs of the high-level modules should be understandable by the meta-architecture. The modules have only their own representations and decision algorithms inside them but the proposed behavior with their associated priority are known by the decision module in order to be able to decompose them into elementary actions.

Another limit can be the lack of reactivity because of the complexity of the meta-architecture. As the role of FlexMex is to take into account all proposed behaviors without inhibitions according the context of the virtual pedestrians, the reactive behaviors will be blended into more complex behaviors or interrupt the current behaviors. The agent does always the more appropriated actions according to its environment and current state. Then sometimes it adapts its behaviors, sometimes not. It depends on its personality, and on the priority of other behaviors. But in any case, reactive behaviors coming from rapid response from the environment will be considered in the decision module but cannot always follow. It is also possible to define a priority on specific behaviors in the decision module, for example for reflex actions that we execute in any cases. However our goal is the credibility of the virtual pedestrians, so we do not wish that reactive behaviors be executed systematically.

7. 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 finalizing work 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, France, within the Terra Dynamica Project (FUI8) supported by CapDigital and Advancity business clusters.

Reference

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111, 1036–1060.
- Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2, 14–23.
- Bryson, J. (2000). Hierarchy and sequence vs. full parallelism in action selection. *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior* (pp. 147–156). MIT Press.
- Campano, S., de Sevin, E., Corruble, V., & Sabouret, N. (2011). Simulating affective behaviours: an approach based on the COR theory. *Proceedings of the Fourth International Conference on Affective Computing and Intelligent Interaction* (pp. 457–466). Memphis, USA.
- de Sevin, E., & Thalmann, D. (2005). A motivational model of action selection for virtual humans. *Proceedings of the Computer Graphics International 2005 Conference* (pp. 213–220).
- Duch, W., Oentaryo, R. J., & Pasquier, M. (2008). Cognitive architectures: Where do we go from here? *Proceedings of the First Conference on Artificial General Intelligence* (pp. 122–136).
- Ferguson, I. A. (1992). Touring Machines: Autonomous agents with attitudes. *Computer*, 25, 51–55.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33, 1–64.

- Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. *Proceedings of the National Conference on Artificial Intelligence*, 21, (pp. 1469–1474).
- Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10, 141–160.
- Maes, P. (1990). Situated agents can have goals. *Robotics and Autonomous Systems*, 6, 49–70.
- Meyer, J. A. (1996). Artificial life and the animat approach to artificial intelligence. In M. Boden (Ed.), *Artificial intelligence* (pp. 325–354). New York: Academic Press.
- Müller, J. P., & Pischel, M. (1993). The agent architecture inteRRaP: Concept and application. *Technical Report* (RR-93-26), German Artificial Intelligence Research Center, Saarbrücken.
- 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).
- Poulet, C., Corruble, V., Seghrouchni, A. E. F., & Ramalho, G. (2011). The open system setting in timed multiagent patrolling. *Proceedings of the 2011 International Conferences on Web Intelligence and Intelligent Agent Technology* (pp. 373–376).
- Rao, A. S., & Georgeff, M. P. (1995). BDI agents: From theory to practice. In V. Lesser (Ed.), *Proceedings of the First International Conference on Multi-Agent Systems* (pp. 312–319).
- Reynaud, Q., de Sevin, E., Donnart, J. Y., & Corruble, V. (2012). A cognitive module in a decision-making architecture for agents in urban simulations. *Proceedings of the Workshop on Cognitive Agents in Virtual Environments (CAVE)*. AAMAS (pp. 1-15). Valencia, Spain.
- Rosenblatt, J. K., & Payton, D. W. (1989). A fine-grained alternative to the subsumption architecture for mobile robot control. *Proceedings of the 1989 International Joint Conference on Neural Networks* (pp. 317–323).
- Rosenblatt, J. K. (1997). DAMN: A Distributed Architecture for Mobile Navigation. *Journal of Experimental Theoretical Artificial Intelligence*, 9, 339–360.
- Schmidt, B. (2005). Human factors in complex systems: The modelling of human behaviour. *Nature*, 4, 5–14.
- Sloman, A. (2001). Varieties of affect and the CogAff architecture schema. *Proceedings of Symposium on Emotion Cognition and Affective Computing AISB* (pp. 39–48).
- Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In R. Sun (Ed.), *Cognition and MultiAgent Interaction* (pp. 79–99).
- Tyrrell, T. (1993). The use of hierarchies for action selection. *Adaptive Behavior*, 1, 387–420.
- Veloso, M., Carbonell, J., Pérez, A., Borrajo, D., Fink, E., & Blythe, J. (1995). Integrating planning and learning: The PRODIGY architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 7(1), 81–120.