



# Rapport du projet IA

---

Sujet : « Emotion Modeling » au niveau  
des agents

**Réaliser par :**

Aouissi Abdessamad

**Année académique : 2020/2021**

## SOMMAIRE

<b>Les systèmes multi-agents</b> .....	3
<b>Définition :</b> .....	3
<b>Le type d'agent</b> .....	3
<b>FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework</b> .....	5
<b>Introduction</b> .....	5
<b>FAtiMA Core :</b> .....	5
<b>FAtiMA Modular</b> .....	10
<b>L'exécution de FAtiMA :</b> .....	12
<b>Conclusion:</b> .....	16
<b>Programming Agents with Emotions</b> .....	17
<b>Introduction:</b> .....	17
<b>2APL and ALMA</b> .....	18
<b>2APL: A Practical Agent Programming Language</b> .....	18
<b>ALMA: A Layered Model of Affect</b> .....	19
<b>Integration of 2APL and ALMA</b> .....	22
<b>Conclusion</b> .....	23
<b>Références</b> .....	24

## Table de figure

Figure 1 : Model d'agent .....	4
Figure 2 : FAtiMA core architecture .....	6
Figure 3 : Processus d'évaluation .....	7
Figure 4 : Pseudocode pour FAtiMA Core .....	9
Figure 5 : La fonction updateEmotions .....	9
Figure 6 : Le cycle de délibération 2APL.....	19
Figure 7 : Les emotion de PAD-space .....	20
Figure 8 : Une représentation graphique de PAD-space .....	20
Figure 9 : Le mappage d'une personnalité dans le modèle Big Five à une valeur PAD.....	20
Figure 10 : La fonction de changement d'humeur push and pull .....	21
Figure 11 : La connexion entre 2APL et ALMA .....	22

## Table des tableaux

Table 1 : Les composants de FAtiMA.....	10
Table 2 Association entre les variables d'évaluation OCC et les types d'émotion OCC .....	11

# Les systèmes multi-agents

## Définition :

Il y a deux approches dans la définition et la conception d'un agent : en isolation et dans le contexte d'une société d'agents (systèmes multi-agents).

Les caractéristiques d'un système multi-agents sont les suivantes :

- Chaque agent a des données ou des possibilités incomplètes pour résoudre un problème,
- Il n'y a pas de système global de contrôle, -
- Les données sont décentralisées, -
- Le calcul est asynchrone.

Un système multi-agents est composé de :

- Un environnement, -
- Un ensemble d'agents, -
- Un ensemble de relations entre les agents, -
- Un ensemble d'opérations

Dans les systèmes multi-agents les agents agissent les uns sur les autres. Ils sont considérés comme des entités autonomes, comme les logiciels ou les robots. Les interactions peuvent être coopératives (si elles ont un objectif commun) ou égoïstes (si elles ont leurs propres objectifs).

## Le type d'agent

Il y a plusieurs types d'agent : **agent réactif**, **agent cognitif**, **agent BDI** et **agent émotionnel**. Dans notre cas nous intéressent qu'avec les agents émotionnels.

Nous devons nous demander si l'intelligence est l'action optimale pour obtenir un but et si un agent intelligent doit avoir seulement un comportement rationnel. Pendant longtemps les processus émotionnels ont été considérés comme indépendants des processus purement cognitifs. Des études récentes ont montré que l'incapacité d'exprimer et ressentir les émotions a des conséquences dans le processus de raisonnement et de décision.

En effet, les émotions, tout comme la personnalité, jouent chez l'homme un rôle extrêmement important dans la perception, la prise de décision et dans les interactions entre les individus.

Une science informatique des émotions est née et elle spécifie que l'agent émotionnel peut faire :

- l'estimation d'une situation comme un événement : la joie, la tristesse,
- La valeur présumée d'une situation qui peut affecter une autre : le ressentiment, la jalousie,
- L'estimation d'une situation comme un événement possible : l'espoir, la crainte,
- L'estimation d'une situation qui confirme ou ne confirme pas une espérance : la satisfaction, la déception

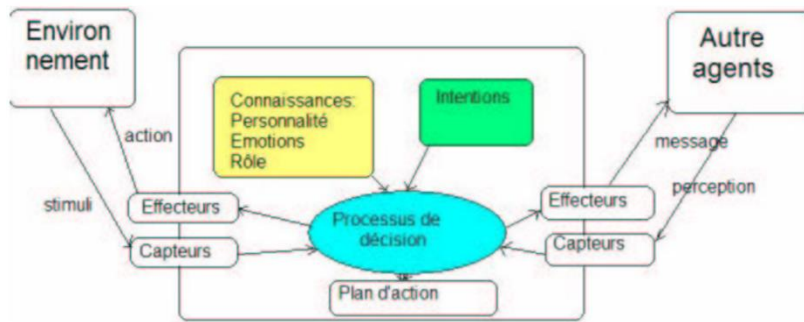


Figure 1 : Model d'agent

Nos agents émotionnels sont des agents intelligents qui ont pour caractéristiques l'autonomie de décision, la capacité de perception et de comportements, la capacité à coopérer, à communiquer, à raisonner et à s'adapter à des nouvelles circonstances.

Les connaissances et les émotions de l'agent sont stockées comme des croyances. Ils se changent avec le temps. En fonction de ces croyances et les désirs de l'agent, il y a un processus de décision, et des intentions et plans sont créés.

Dans ce qui suit nous avons voir la modélisation des agent emotionele avec l'architecture FATiMA Modular et aussi avec un langage de programation des agents dédiée a la programmation des agents avec émotion.

# FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework

## Introduction

Afin de modéliser les comportements émotionnels des agents, (Dias, Mascarenhas, & Paiva, 2011) ont proposé FAtiMA – Fearnot AffecTive Mind Architecture). Cette architecture a été étendue dans plusieurs travaux de recherche pour simuler différents phénomènes émotionnels pour les agents virtuels. Comme ces utilisations sont de différentes natures, l'implémentation concrète de l'architecture est différée d'un travail à un autre (FearNot! (Paiva, et al., 2005), ORIENT (Ruth, Vannini, Andre, Paiva, Enz, & Hall, 2009), Model of Empathy (Rodrigues, Mascarenhas, Dias, & Paiva, 2009), Cultural Behaviour (Mascarenhas, Dias, Prada, & Paiva, 2010), Drives (Lim, Dias, Ruth, & Paiva, 2011)). L'architecture résultante peut être difficile à utiliser car la complexité de la compréhension de l'architecture augmente avec le nombre de fonctionnalités présentes. Pour cette raison, beaucoup d'efforts ont été consacrés à la création d'une version modulaire de l'architecture, dans laquelle les fonctions et les opérations sont divisées en composants modulaires indépendants. Cela nous permet d'utiliser des versions plus légères et plus simples de FAtiMA avec seulement certains composants (par exemple, des composants réactifs et émotionnels uniquement).

Lors de la refactorisation, les processus les plus pertinents ont été généralisés rendant l'architecture plus facile à étendre en nous permettant de travailler indépendamment sur différents composants. L'un des processus les plus pertinents généralisés dans l'architecture était le processus d'évaluation. Le raisonnement suivi était que l'architecture devrait être capable d'incorporer plus tard plusieurs mécanismes d'évaluation distincts et même des théories d'évaluation. Il existe actuellement plusieurs théories d'évaluation différentes qui modélisent le processus de génération d'émotions (OCC [9], Roseman [13], Scherer [15]).

L'objectif de ce chapitre est un premier pas vers une contribution aux standards de la modélisation émotionnelle ; il s'agit donc de présenter une architecture de base (core architecture) générique pour les agents émotionnels, et de décrire comment le mécanisme d'évaluation générique pourrait être utilisé pour mettre en œuvre différentes théories d'évaluation.

FAtiMA Modular est composé d'une couche centrale nommée FAtiMA Core.

## FAtiMA Core :

FAtiMA Core est un modèle qui définit généralement le fonctionnement de l'architecture de l'agent. FAtiMA Core modélise le processus émotionnel en deux étapes : Dérivation Cognitive (Appraisal Derivation en anglais), et Dérivation Affective (Affect Derivation en anglais). La Dérivation Cognitive est chargée d'évaluer la pertinence de l'événement pour l'agent et de déterminer un ensemble de variables d'évaluation (par exemple, désirabilité et désirabilité pour les autres dans le cas de l'OCC). La deuxième étape, la dérivation d'affect, prend les variables d'évaluation en entrée et génère les états affectifs résultants (émotions ou humeur) selon une théorie de l'évaluation, peut se baser aussi sur le modèle OCC ou de Scherer. Par exemple, si le niveau de désirabilité d'un événement est élevé, la joie peut être reproduite.

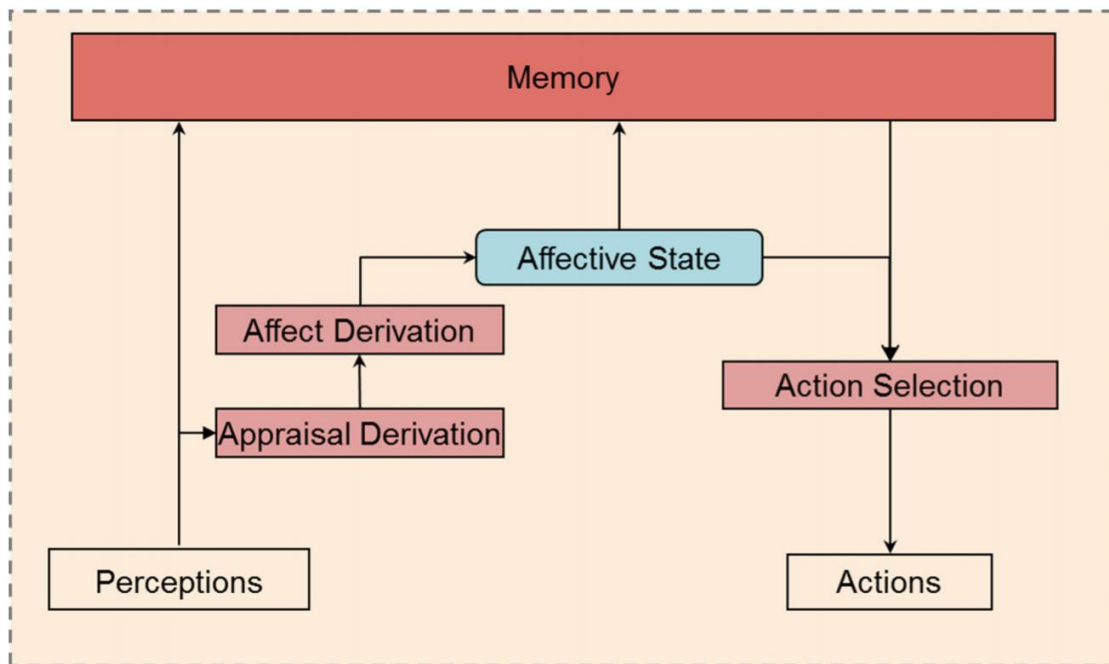


Figure 2 : FATiMA core architecture

La figure 2 montre un schéma de FATiMA Core avec les fonctionnalités de base pour une architecture d'agent émotionnel. Un agent est capable de recevoir des perceptions de l'environnement (events) qui sont utilisées pour mettre à jour la mémoire (memory) de l'agent et déclencher le processus d'évaluation (appraisal process). Le résultat du processus d'évaluation est stocké dans l'état affectif (affective state), et utilisé plus tard pour influencer les processus de sélection d'action (action selection) qui feront agir l'agent sur l'environnement.

Tous les composants sont conçus selon deux propriétés principales : ils doivent être interchangeables, c'est-à-dire pouvoir être remplacés, ajoutés ou retirés avec un minimum d'effort ; et ils doivent être faiblement couplés - les dépendances entre les composants doivent être évitées à moins que cela ne soit strictement nécessaire.

### Processus d'évaluation (appraisal process) :

L'un des principaux objectifs lors de la conception du mécanisme d'évaluation était qu'il devait être suffisamment puissant et flexible pour représenter la plupart des théories d'évaluation.

Selon la théorie d'évaluation Scherer, l'évaluation se fait à plusieurs niveaux, niveau sensoriel, niveau schématique et niveau conceptuel. Les niveaux inférieurs sont généralement plus rapides pour identifier un contrôle d'évaluation incitatif, mais fournissent souvent des informations d'évaluation moins que parfaites. Dans les cas où les évaluations de niveau inférieur ne peuvent pas évaluer correctement la situation, des évaluations de niveau supérieur sont effectuées mais plus complexes et plus lourdes.

De plus, afin de rendre le processus d'évaluation aussi flexible que possible, l'évaluation FATiMA est fondée sur un ensemble de principes de conception, qui tentent de généraliser les exigences de Scherer pour sa théorie de l'évaluation :

1- L'évaluation est progressive, c'est-à-dire que l'évaluation n'est pas un processus unique et différentes variables d'évaluation et les émotions correspondantes peuvent être générées à différents moments. Par conséquent, le processus d'évaluation doit être exécuté en continu.

2- Un composant d'évaluation peut dépendre d'une variable d'évaluation déterminée par un autre composant et doit donc accéder à toutes les variables d'évaluation générées dans le processus d'évaluation. Cela implique que nous avons besoin d'un moyen de stocker le résultat d'un composant d'évaluation (une structure semblable aux registres) et de l'utiliser comme entrée pour d'autres composants d'évaluation.

3- Lazare a souligné que l'évaluation est souvent suivie d'une réévaluation, dans le but de corriger l'évaluation sur la base de nouvelles informations ou d'un traitement plus approfondi. Par conséquent, la valeur d'une variable d'évaluation peut changer avec le temps, et l'état affectif doit refléter ce changement. Cela signifie que l'architecture Core doit imposer que les processus de dérivation d'effet soient exécutés chaque fois qu'un nouveau résultat est renvoyé par les composants d'évaluation.

4- N'importe quel composant peut contribuer à n'importe quelle variable d'évaluation. Pour cette raison, nous devons stocker la contribution de toutes les composantes à une variable d'évaluation donnée. Afin de déterminer la valeur finale d'une variable d'évaluation, une politique explicite est utilisée. La politique peut choisir la valeur la plus élevée, effectuer une somme pondérée, utiliser la valeur déterminée par le dernier composant pour définir la variable ou utiliser un mécanisme de priorité.

La figure 3 présente un schéma du processus d'évaluation, un événement généré déclenchera le début d'un processus d'évaluation créant un cadre d'évaluation (Appraisal Frame).

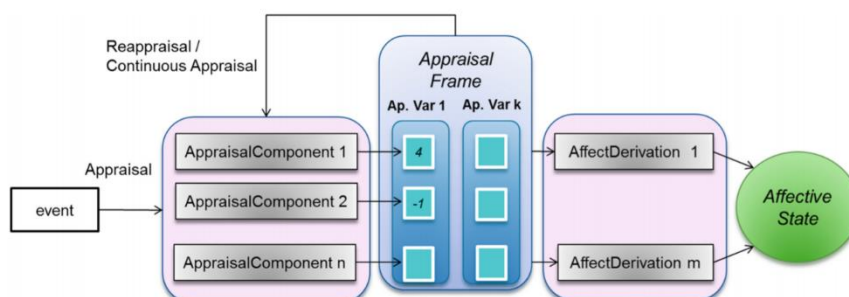


Figure 3 : Processus d'évaluation

Dans l'exemple fourni, l'une des composantes définit initialement la variable d'évaluation avec une valeur positive (4), mais une autre composante la définit ultérieurement comme étant légèrement négative (-1). La flèche du cadre d'évaluation vers les composants d'évaluation représente que l'évaluation peut prendre plusieurs cycles et dépend de l'évaluation d'autres composants.

Chaque fois qu'il y a un changement dans un cadre d'évaluation, l'état émotionnel de l'agent doit refléter ce changement. Ceci est géré par les composants Affect Derivation, dont la seule responsabilité est de générer et de mettre à jour les émotions de l'agent. Chaque composant de dérivation d'affect est informé des changements dans les variables d'évaluation stockées dans le cadre d'évaluation. Ces informations sont ensuite utilisées pour décider quelles émotions créer et leur intensité correspondante. Des facteurs de modulation supplémentaires, tels qu'une personnalité ou un préjugé culturel, peuvent également être utilisés par le composant pour déterminer l'intensité de l'émotion. Alors, plusieurs émotions peuvent être ajoutées simultanément à l'état affectif de l'agent, ce qui entraîne alors pour l'agent des émotions mixtes, par conséquent l'état affectif de l'agent intègre également la notion d'humeur. L'humeur représente un état affectif global qui est influencé par les émotions vécues par l'agent, les émotions positives augmentent l'humeur, tandis que les émotions négatives la diminuent.

Un dernier mécanisme important est la manière dont les émotions sont stockées et organisées dans l'état affectif, les émotions sont indexées dans l'état affectif par l'ensemble des variables d'évaluation utilisées pour les générer ainsi que l'événement correspondant. Lorsqu'une nouvelle émotion est ajoutée à l'état émotionnel, l'état émotionnel vérifie si une émotion causée par le même événement et déclenchée par exactement les mêmes variables d'évaluation existe déjà. Si c'est le cas, alors il remplacera l'émotion existante par la nouvelle. Sinon, cela ajoutera simplement la nouvelle émotion. Notez qu'en utilisant ce mécanisme, il est possible de générer de nombreuses émotions distinctes associées au même événement tant qu'elles sont déclenchées par un ensemble de variables différent.



```

while(shutdown != true)
  for each Component c
    c.update();

  e <- perceiveEvent();

  if(a new event is perceived)
    memory.update(e);
    for each Component c
      c.update(e);

  aF <- newAppraisalFrame(e);

  for each AppraisalComponent aC
    aC.startAppraisal(e, aF);
    updateEmotions(aF);

  for each AppraisalComponent aC
    aF <- aC.continuousAppraisal();
    updateEmotions(aF);

  for each BehaviorComponent bC
    bc.actionSelection();

  a <- selectAction();
  executeAction(a);

```

*Figure 4 : Pseudocode pour FAtiMA Core*

La figure 4 présente le pseudocode pour FAtiMA Core. La première fonction de mise à jour est utilisée pour mettre à jour les composants à chaque cycle. Ensuite, le processus perceptif vérifie s'il y a une nouvelle perception dans l'environnement. Si un nouvel événement est perçu, il est utilisé pour mettre à jour la mémoire de l'agent et tous les composants existants et un nouveau cadre d'évaluation sera créé associé à l'événement. Le cadre d'évaluation sera ensuite utilisé pour démarrer le processus d'évaluation. Une fois le cadre d'évaluation déterminé, la fonction `updateEmotions` sera utilisée pour lancer le processus `AffectDerivation`.

```

updateEmotions(aF)

if(aF.hasChanged())
  for each AffectDerivationComponent aD
    emotion <- aD.affectDerivation(aF)
    AffectiveState.add(emotion);

```

*Figure 5 : La fonction `updateEmotions`*

Selon la figure 5, chaque fois qu'un changement est détecté dans le cadre d'évaluation, tous les composants de dérivation d'évaluation existants sont utilisés pour générer les émotions correspondantes, qui sont ajoutées à l'état émotionnel.

Enfin, à la fin du cycle d'un agent, le Core demandera aux composantes du comportement (behaviour components) s'ils souhaitent effectuer une action. Il peut s'agir d'un simple mécanisme réactif ou d'un mécanisme délibératif complexe. Si plus d'une action est déclenchée par différents composants, un mécanisme de priorité sera utilisé pour sélectionner le plus pertinent. L'action sélectionnée sera alors exécutée dans l'environnement.

## FAtiMA Modular

Un agent FAtiMA qui n'a qu'un Core ne fera rien. L'architecture modulaire FAtiMA est créée en ajoutant un ensemble de composants au Core. L'architecture FAtiMA a été définie avec les composants suivants :

Deliberative Component	Délibère sur quel objectif doit être poursuivi à un moment donné et quels plans l'agent doit-il suivre pour l'atteindre. Générer des émotions liées aux objectifs. Ces variables d'évaluation sont GoalStatus, GoalConduciveness et GoalSuccessProbability.
Reactive Component	Ce composant utilise des règles de réaction émotionnelle prédéfinies pour déterminer la valeur des variables d'évaluation OCC suivantes : Desirability, DesirabilityForOthers, Praiseworthiness et Like
OCCAffectDerivation Component	Génère des émotions à partir des variables d'évaluation selon la théorie des émotions OCC (voir table 2)
Motivational Component	Intègre cinq besoins humains différents chez les agents : énergie, intégrité, affiliation, certitude et compétence.
Theory of Mind Component	Capacité à attribuer des états mentaux (croyances, désirs, émotions et intentions) à d'autres agents et à raisonner sur ces états.
Cultural Component	Il met en œuvre le comportement dépendant de la culture des agents à travers l'utilisation de rituels, de symboles et de dimensions culturelles.

*Table 1 : Les composants de FAtiMA*

Pour illustrer comment ces plusieurs composants sont combinés dans le scénario pour créer un processus d'évaluation complexe, considérons l'exemple suivant. Lors d'un dîner organisé par cinq agents différents, l'un des agents informe les autres qu'il se sent malade. Lorsque la

culture de l'agent est définie comme hautement collectiviste, il y a un agent qui décide d'offrir sa propre médecine. Après avoir reçu le médicament, l'agent malade suscite des émotions de joie et d'admiration.

Le processus d'évaluation qui a lieu pour l'agent destinataire se déroule de la manière suivante. Premièrement, la composante de l'état de motivation déterminera rapidement que la réception du médicament aura un effet positif sur l'intégrité de l'agent (qui est faible pour le moment) générant une forte désirabilité positive, et stockera cette valeur dans le cadre d'évaluation. Le noyau FAtiMA détectera un changement dans le cadre d'évaluation et initiera le processus AffectDerivation en appelant le composant OCCAffectDerivation. Cela conduit à la création initiale d'une émotion de joie.

Appraisal variables	Associated emotion types
Desirability	Joy, Distress
Desirability, DesirabilityForOthers	HappyFor, Gloating, Pitty, Resentment
Praiseworthiness	Pride, Admiration, Shame, Reproach
Praiseworthiness, Desirability	Gratification, Gratitude, Remorse, Anger
Like	Love, Hate
GoalStatus, GoalConduciveness, GoalSuccessProbability	Hope, Fear, Relief, Satisfaction, Fears-Confirmed, Disappointment

Table 2 Association entre les variables d'évaluation OCC et les types d'émotion OCC

Quelques cycles plus tard, le composant Théorie de l'esprit déterminera également que le même événement aura un effet négatif sur la volonté d'intégrité de l'agent donneur (puisqu'il a perdu son médicament), déterminant le même événement comme indésirable pour le donneur. AffectDerivation sera appelé à nouveau, mais cette fois aucune émotion n'est générée. Étant donné que la composante culturelle nécessite les deux autres variables d'évaluation pour générer l'évaluation, elle ne se déclenchera pas tant que les deux valeurs ne seront pas déterminées par les autres composantes.

Dès lors, la composante culturelle détermine la variable de louange (praiseworthiness variable). Dans cet exemple, compte tenu de la culture collectiviste de l'évaluateur, il jugera l'événement très louable. Le composant OCCAffectDerivation va alors générer une émotion Admiration basée sur la variable praiseworthiness.

Une fois qu'une variable d'évaluation est créée, ses valeurs sont utilisées pour générer une émotion selon les règles suivantes :

Pour la variable Desirability :

Positive	Negative
Joy	Distress

Pour la variable Like :

Positive	Negative
Love	Hate

Pour la variable Praiseworthiness :

Praiseworthiness	Objet de l'action	
	Self	Other
Positive	Pride	Admiration
Negative	Shame	Reproach

Pour les variables Desirability et Desirability for Other :

Desirability	Desirability for Other	
	Positive	Negative
Positive	Happy-For	Gloating
Negative	Resentment	Pitty

Pour les variables Praiseworthiness et Desirability :

Émotions de base	Émotion composée
Joy + Pride	Gratification
Joy + Admiration	Gratitude
Distress + Shame	Remorse
Distress + Reproach	Anger

## L'exécution de FAtiMA :

On exécute un simulateur simple de l'architecture FAtiMA pour tester et déboguer des agents et simuler des interactions simples.

```
> AgentLauncher
> data
> FAtiMA
> FAtiMA.AdvancedMemory
> FAtiMA.culture
> FAtiMA.DeliberativeComponent
> FAtiMA.EmotionalIntelligence
> FAtiMA.Empathy
> FAtiMA.MotivationalSystem
> FAtiMA.OCCAffectDerivation
> FAtiMA.ReactiveComponent
> FAtiMA.SocialRelations
> FAtiMA.ToM
> LanguageServer
> WorldTest
```

On utilise trois composant de base ReactiveComponent, OCCAffectDerivation Component et Deliberative Component, il est possible d'ajouter n'importe quel composant de FAtiMA.

```
AgentCore aG = initializeAgentCore(args);
ArrayList<String> extraFiles = new ArrayList<String>();
String cultureFile = ConfigurationManager.getMindPath() + ConfigurationManager.getOptionalConfiguration();
String relationsFile = ConfigurationManager.getMindPath() + "relations.xml"; //fixed for now, read it as

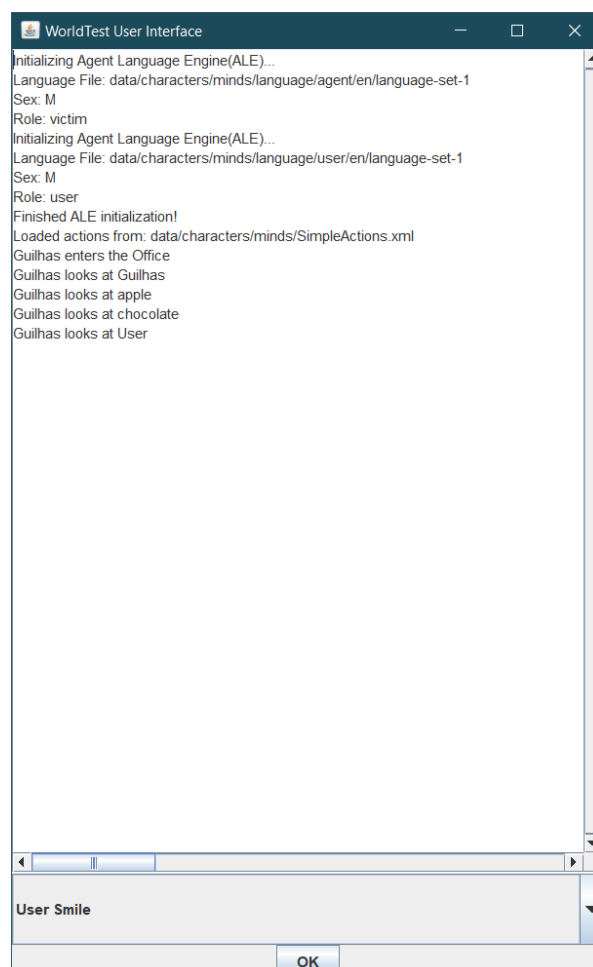
if (!aG.getAgentLoad())
{
    extraFiles.add(cultureFile);

    //FAtiMA Light
    aG.addComponent(new ReactiveComponent());
    aG.addComponent(new OCCAffectDerivationComponent());
    aG.addComponent(new DeliberativeComponent());

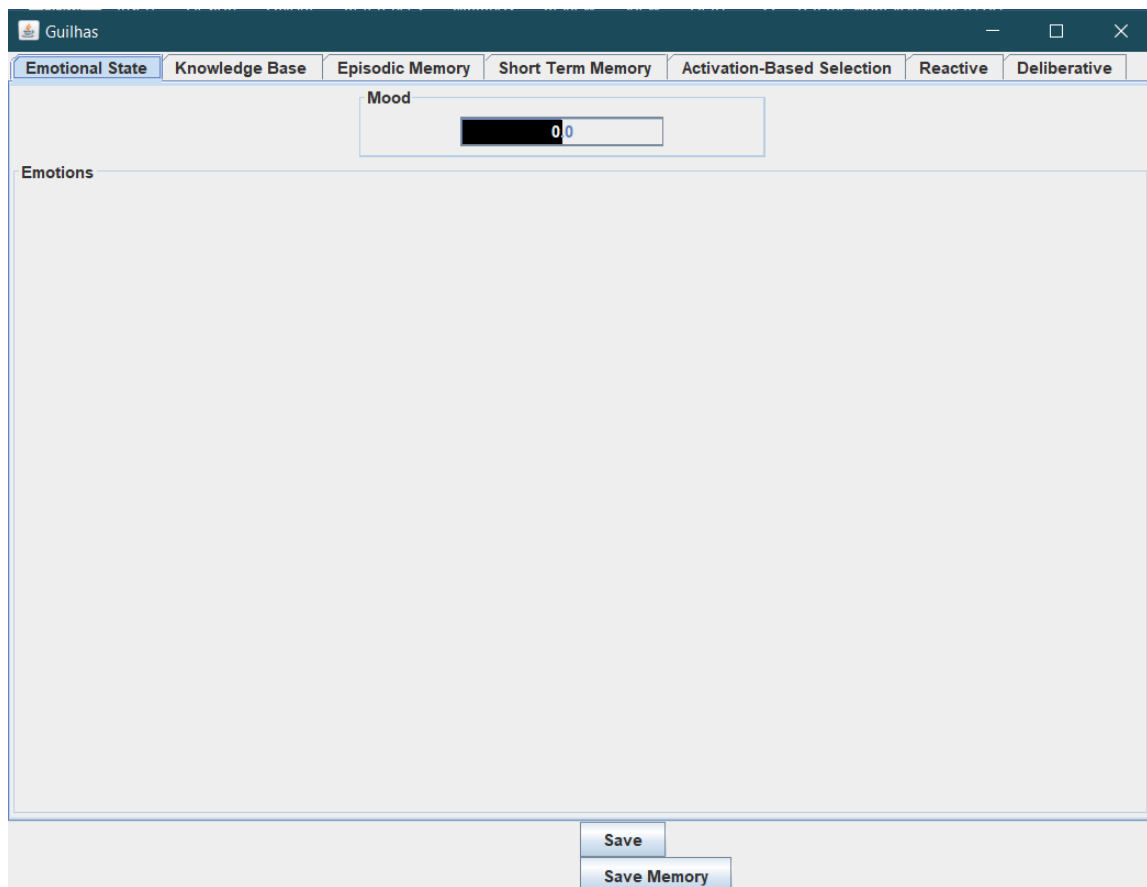
    //FAtiMA Advanced Components
    //aG.addComponent(new MotivationalComponent(extraFiles));
    //aG.addComponent(new SocialRelationsComponent(relationsFile, extraFiles));
    //aG.addComponent(new ToMComponent(ConfigurationManager.getName()));
    //aG.addComponent(new CulturalDimensionsComponent(cultureFile));
    //aG.addComponent(new AdvancedMemoryComponent());
    //aG.addComponent(new EmotionalIntelligenceComponent(extraFiles));

}
aG.StartAgent();
```

Pour exécuter FAtiMA il doit lancer les deux programmes AgentLauncher et WorldTest.

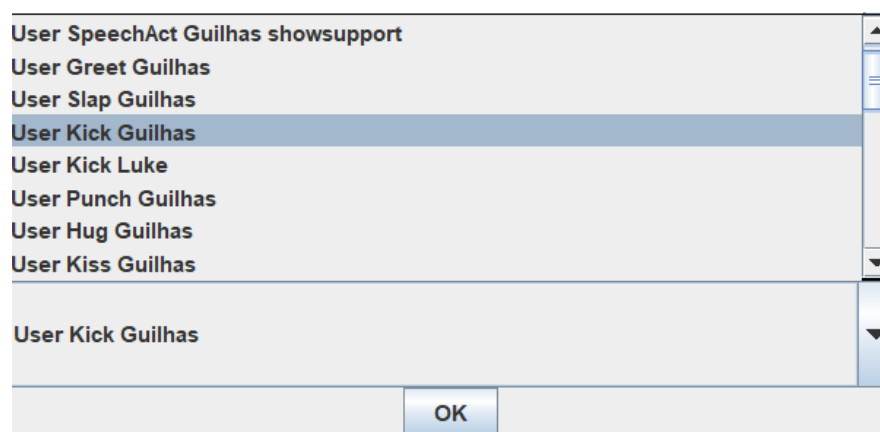


AgentLauncher est lancer avec le parametre Guilhas.



L'agent Guilhas apparaît avec une humeur générale neutre, lors de la perception des événements on s'aperçoit que des émotions sont générés et l'humeur prend aussi une valeur positive ou négative selon les émotions qui se forment.

Voici exemple des évènements :

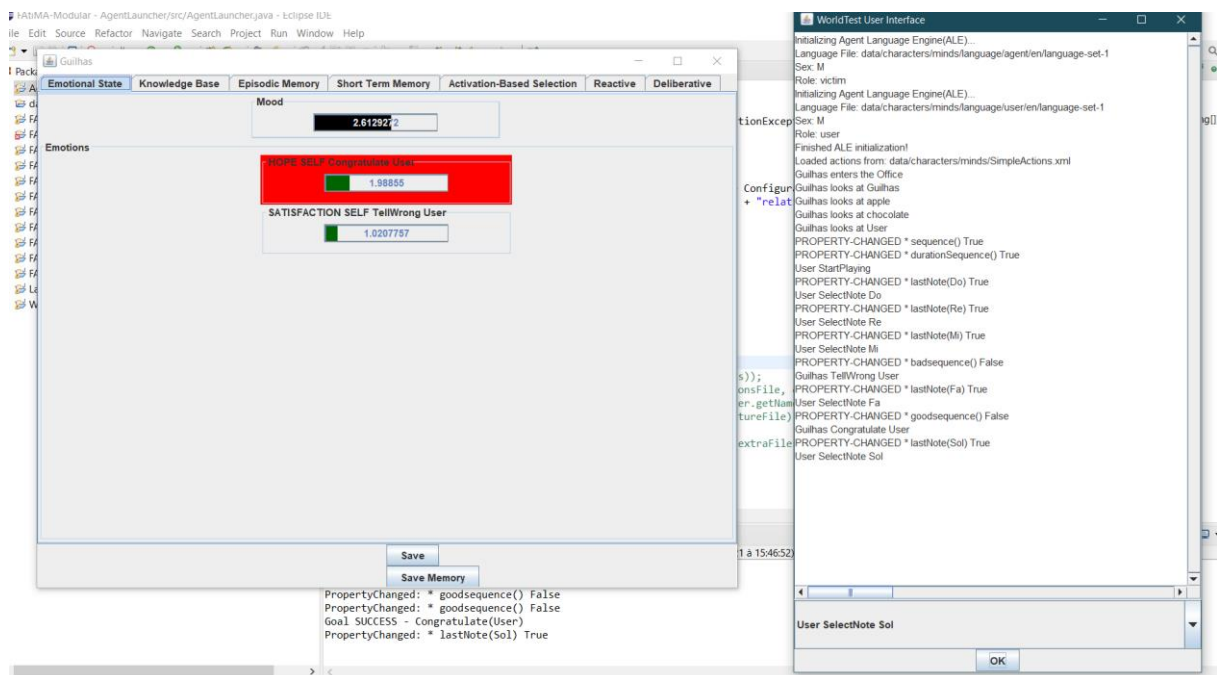


Après lance les évènements suivants :

User StartPlaying

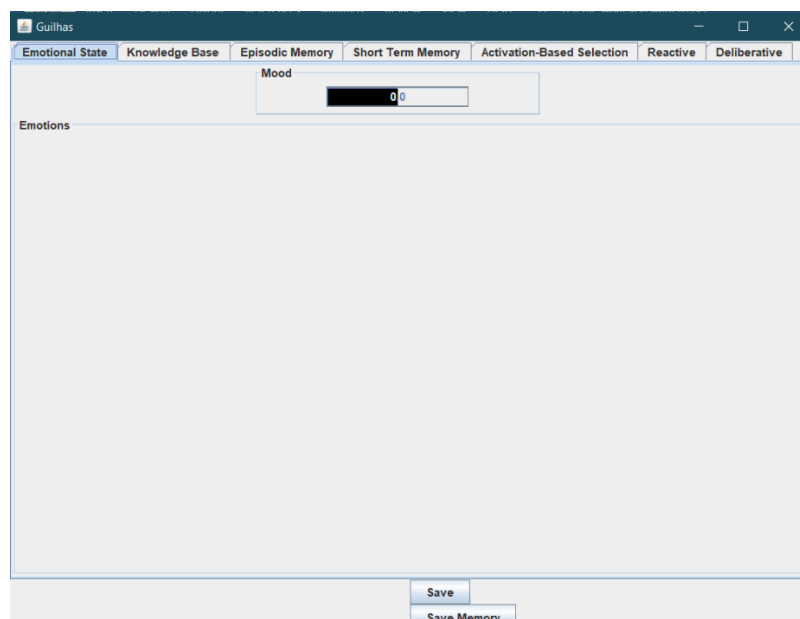
User SelectNote Do

User SelectNote Re  
 User SelectNote Mi  
 User SelectNote Fa  
 User SelectNote Sol



L'agent commence à jouer au piano et frappera les note Do, Re et Mi, L'émotion Hope est générée, et lorsqu'il continuera frappe les note Fa et Sol, l'émotion de Satisfaction aussi apparaitre. L'humeur augmente aussi

Lorsque l'agent est laissé pendant un certain temps sans qu'un événement se déclenche, les émotions commencent à s'estomper et l'humeur devient neutre.



## **Conclusion:**

FAtiMA Modular est une architecture générique et flexible pour les agents émotionnels, avec ce que nous considérons comme l'ensemble minimum de fonctionnalités qui nous permet de mettre en œuvre et de comparer différentes théories d'évaluation dans un scénario donné. Elle est composée d'un algorithme de base et d'un ensemble de composants qui ajoutent des fonctionnalités particulières (que ce soit en termes d'évaluation ou de comportement) à l'architecture, ce qui rend l'architecture plus flexible et plus facile à étendre.



# Programming Agents with Emotions

## Introduction:

Une caractéristique distinctive d'un agent est son autonomie, c'est-à-dire sa capacité à décider et à exécuter des actions. Cette fonctionnalité nécessite qu'un agent ait des objectifs ou des préférences concernant l'état de son environnement (y compris lui-même et d'autres agents) et des croyances concernant l'état actuel de son environnement également. La théorie de la décision rationnelle informe un agent de sélectionner les actions qui sont censées atteindre, ou maximiser les chances d'atteindre ses objectifs, la théorie de la décision rationnelle préconise de maximiser la réalisation des objectifs, est soutenu qu'au moins une partie de l'écart entre le comportement tel que prescrit par la théorie de la décision rationnelle et celui observé chez les agents humains et dû au rôle des émotions dans la prise de décision humaine.

Que les émotions influencent le comportement des humains est immédiatement clair dans le cas de la détresse. Lorsqu'ils sont en détresse, les agents humains se comportent différemment et prennent généralement d'autres mesures qu'ils ne le feraient autrement. En psychologie et en sciences cognitives, un certain nombre de théories des émotions traitant de plusieurs aspects des émotions ont été proposées. Ces aspects concernent l'appréciation des événements, l'expérience des émotions et la gestion des émotions (également appelée régulation des émotions). Le but de la régulation des émotions est de maintenir les émotions désirables et de diminuer les émotions indésirables. Ainsi, la régulation des émotions a un effet direct sur le comportement de l'agent.

Il existe de nombreuses propositions pour modéliser et concevoir des agents artificiels basés sur la théorie de la décision rationnelle. le BDI est un modèle de décision qualitatif qui explique les décisions rationnelles d'un agent en termes d'informations de l'agent sur l'état actuel du monde (croyance), les états que l'agent veut atteindre (désir) et ses engagements envers des choix (Intention). Le modèle BDI s'est avéré être un modèle efficace pour la planification réactive et pour les agents ayant des objectifs complexes interagissant avec des environnements hautement dynamiques.

Afin de construire des agents logiciels rationnels, plusieurs langages de programmation d'agents ont été développés. Ces langages, souvent appelés langages de programmation d'agents basés sur BDI, fournissent des constructions de programmation pour définir les agents en termes d'états mentaux, d'actions, d'événements et de plans. L'interpréteur de ces langages de programmation contient un composant de prise de décision qui sélectionne et exécute en permanence les actions qui permettent d'atteindre au mieux les objectifs de l'agent ou de maximiser ses préférences.

Dans ce chapitre, nous illustrons l'intégration d'un modèle d'émotion dans des agents BDI en utilisant le langage de programmation d'agents basé sur BDI 2APL (A Practical Agent Programming Language) et le modèle de génération d'émotion et d'humeur ALMA (A Layered Model of Affect).

## 2APL and ALMA

Avant d'expliquer 2APL, ALMA et leur intégration, d'autres Framework a été examine tels que EMA et Cathexis. Après avoir étudié ces cadres, il arrivés à la conclusion que ALMA était le meilleur candidat pour étendre le 2APL avec des émotions. Car, ALMA est basé sur une théorie solide et l'explication du modèle est claire. Nous pensons que ALMA fait un assez bon travail pour simuler la génération d'émotions et d'humeurs. Une dernière raison pour laquelle nous avons choisi ALMA est la facilité d'intégration avec l'approche basée sur les règles de 2APL.

## 2APL: A Practical Agent Programming Language

2APL a été développé pour prendre en charge la programmation de systèmes multi-agents qui sont considérés comme constitués d'agents autonomes qui interagissent les uns avec les autres et avec leur environnement.

Un agent est défini en termes de **croiances** (représentées comme un programme Prolog), **d'objectifs** (chacun représenté comme une conjonction d'atomes), **d'intentions** (représentées comme des plans constitués d'actions), d'un ensemble **d'actions internes** (mises à jour des croyances et des objectifs, croyance et tests d'objectifs) et les **actions externes** (par exemple, les actions à effectuer dans l'environnement environnant, y compris les actions de communication), et **trois ensembles de règles de raisonnement** pratiques qui génèrent des plans lorsqu'ils sont appliqués.

Le premier type de règles est conçu pour générer des plans pour atteindre des objectifs (appelées règles d'objectif de planification, ou règles PG), le second pour traiter des événements externes, des messages et des actions abstraites (appelées règles d'appel de procédure, ou règles PC), et le troisième pour réparer les plans échoués (règles de réparation de plan, ou règles PR).

Les règles de raisonnement pratique en 2APL ont la forme générale suivante :

**{head} ← {belief query} | {body}**

- **Les règles PG** sont utilisées pour générer des plans pour atteindre des objectifs spécifiques.

**beHome() ← hours(X) and X > 17 | { $\pi$ }**

Cette règle indique que le plan  $\pi$  doit être généré et ajouté aux intentions de l'agent si ce dernier a le but beHome() (être à la maison) et pense que l'heure actuelle est postérieure à 17 heures.

- **Les règles PC** : la tête de la règle est un événement et son corps consiste en un plan.

**event(rainStarted(), env) ← noUmbrella | { $\pi$ }**

Cette règle indique qu'un plan  $\pi$  doit être généré chaque fois que l'événement rainStarted() est reçu de l'environnement env et que l'agent pense ne pas avoir de parapluie.

- **Les règles PR** : La tête et le corps de ce type de règle sont constitués de plans.

**takeBus;  $\pi$  ← haveMoney | {takeTaxi;  $\pi$ }**

Cette règle indique que lorsqu'un plan commençant par l'action takeBus (prendre un bus) échoue (le plan échoue parce que l'exécution de sa première action échoue) et que l'agent

pense avoir assez d'argent, alors l'agent peut réparer son plan en remplaçant l'action takeBus par l'action takeTaxi (prendre un taxi).

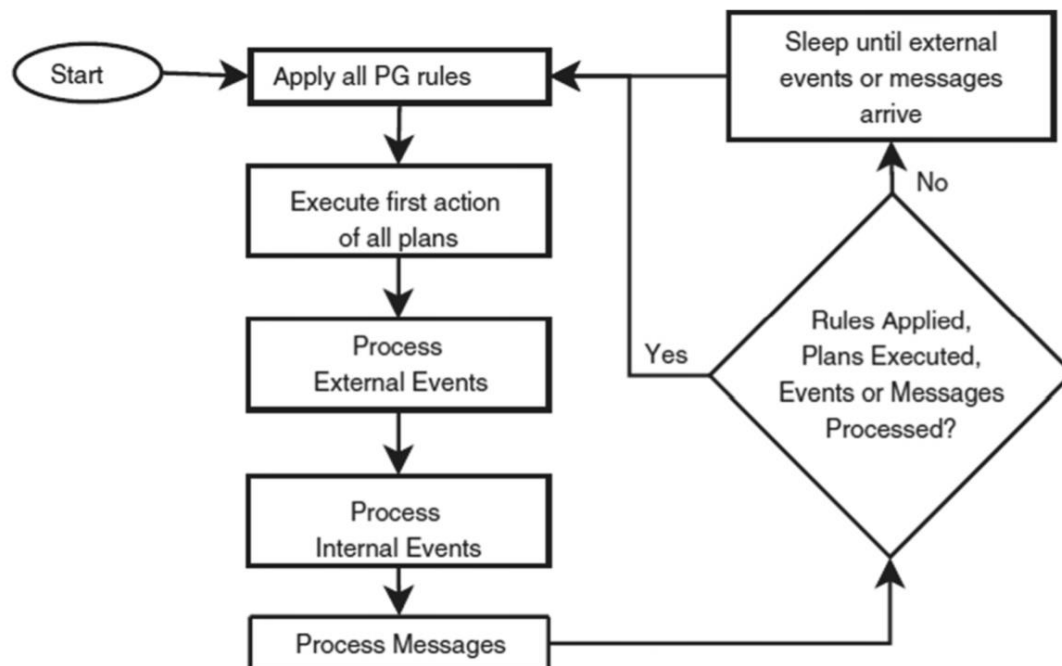


Figure 6 : Le cycle de délibération 2APL

L'exécution des programmes d'agents 2APL est réalisée par l'interpréteur 2APL qui met en œuvre le cycle de délibération illustré à la Fig. 6. Le cycle de délibération commence par l'application des règles PG applicables d'un programme d'agent pour générer des plans pour atteindre les objectifs de l'agent. Le cycle de raisonnement se poursuit par l'exécution des plans générés. Ensuite, les événements et les messages internes et externes reçus sont traités en appliquant les règles PC et PR.

## ALMA: A Layered Model of Affect

Le modèle ALMA considère l'état affectif de l'agent en trois niveaux : **émotion**, **humeur** et **personnalité**. L'**émotion** est l'état affectif à court terme et qui a pour objet de répondre à un événement, une action ou un objet. L'**humeur** est l'état affectif à moyen terme qui ne se rattache ni à un événement, ni à une action, ni à un objet spécifique. L'humeur est stable pendant une certaine durée de temps (généralement prédéfinie par le concepteur) et est influencée par le résultat d'analyse cognitive de l'agent (i.e. l'émotion ressentie lors de la perception d'un événement). L'humeur dans ce modèle est définie comme la moyenne de l'état émotionnel de l'agent à travers diverses situations. La **personnalité** est l'état affectif à long terme. Elle représente les caractéristiques mentales de l'agent et détermine comment l'émotion de l'agent va évoluer en fonction de l'information perçue et de son humeur. Le modèle ALMA est basé sur **EmotionEngine** qui est une implémentation du modèle d'émotion OCC augmenté du modèle Big Five des traits de personnalité. **EmotionEngine** correspondant calcule les émotions et les humeurs selon un modèle d'émotion proposé par Mehrabian, dans EmotionEngine, il existe 24 émotions différentes. **ALMA** utilise le modèle de personnalité **Big Five** (proposé par (McCrae & John, 1992)) pour les agents. Ce modèle permet de définir

les traits de personnalité en fonction de 5 dimensions : Extraversion, Névrose, Agréabilité, Conscience, Ouverture à l'expérience.

L'émotion de l'agent envers un événement est déterminée en utilisant le modèle OCC.

L'intensité de cette émotion est calculée en se basant sur le trait de personnalité. Puis

l'humeur va être modifiée en fonction de l'émotion ressentie.

Pour la représentation des émotions, l'auteur utilise la représentation dimensionnelle avec trois dimensions nommée **PAD-space** : Valence (**Pleasure**), Activation (**Arousal**), et Dominance (**Dominance**) dont chacune a une valeur comprise entre -1 et 1.

+P+A+D Exuberant	-P-A-D Bored
+P+A-D Dependent	-P-A+D Disdainful
+P-A+D Relaxed	-P+A-D Anxious
+P-A-D Docile	-P+A+D Hostile

Figure 7 : Les emotion de PAD-space

La figure 8 montre une représentation graphique de l'espace PAD d'un individu à un moment donné. L'octant en surbrillance montre l'humeur discrète actuelle et la grande sphère (verte) représente l'humeur actuelle, y compris son intensité. Les sphères plus petites (rouges)

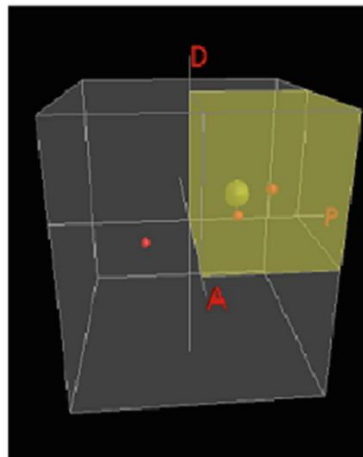


Figure 8 : Une représentation graphique de PAD-space

montrent les émotions actuellement actives. Notez qu'ALMA couvre l'évaluation de la génération d'émotions et d'humeurs, mais ne couvre pas l'adaptation.

Les personnalités peuvent être mappées dans l'espace PAD, comme le montre la figure 9. Les facteurs Big Five sont mappés sur les valeurs PAD, qui décrivent l'humeur par défaut de l'agent.

$$\begin{aligned}
 \text{Pleasure} &:= 0.21 * \text{Extraversion} + 0.59 * \text{Agreeableness} + 0.19 * \text{Neuroticism} \\
 \text{Arousal} &:= 0.15 * \text{Openness} + 0.30 * \text{Agreeableness} - 0.57 * \text{Neuroticism} \\
 \text{Dominance} &:= 0.25 * \text{Openness} + 0.17 * \text{Conscientiousness} + 0.60 * \text{Extraversion} - \\
 &\quad 32 * \text{Agreeableness}
 \end{aligned}$$

Figure 9 : Le mappage d'une personnalité dans le modèle Big Five à une valeur PAD

Dans ALMA, un document AffectML (variation sur XML) doit être spécifié pour chaque agent. Ce document contient les paramètres de calcul globaux du modèle d'agent et un profil de personnalité. Le profil de personnalité d'un agent contient une balise qui décrit la personnalité de l'agent en fonction des cinq facteurs du modèle Big Five, ainsi que des balises qui décrivent ce que l'on appelle les « règles d'évaluation ». Ces règles décrivent comment un agent évalue son environnement, c'est-à-dire comment il évalue les actions, les objets et les événements, y compris les affichages d'émotion et d'humeur. Les affichages d'émotions et d'humeurs sont respectivement les aspects visuels des émotions et des humeurs ; par exemple, une larme peut faire partie d'un affichage d'émotion pour l'émotion tristesse. En général, les règles d'évaluation se composent d'une partie d'appariement et d'un corps d'action. La partie correspondante peut décrire une action, un objet ou un événement. Elle est comparée aux entrées des scripts ALMA (notez qu'ALMA fonctionne avec des scripts, alors que 2APL utilise des programmes). Lorsqu'un affichage d'événement, d'action, d'objet, d'émotion ou d'humeur se produit dans le script qui correspond à la partie correspondante d'une règle d'évaluation, le corps d'action de cette règle est exécuté. Le corps de l'action peut être une ou plusieurs « étiquettes d'évaluation » ou des conditions génératrices d'émotions (EEC). Les EEC sont des entités qui provoquent la génération d'émotions selon le modèle OCC, les EEC utilisés dans ALMA sont la louange (des actions), la désirabilité (des événements), l'attractivité (des objets) et la sympathie (d'une personne). Les EEC sont des entrées directes dans EmotionEngine, tandis que les balises d'évaluation doivent être converties en EEC avant de pouvoir être utilisées comme entrée pour EmotionEngine, EmotionEngine va alors générer de nouvelles émotions (et mettre à jour les intensités d'émotions déjà existantes) et, en fonction des émotions, il va mettre à jour l'humeur de l'agent. Les émotions sont représentées par un point dans l'espace PAD. Parallèlement à ce point vient également une intensité de l'émotion qui se dégrade au fil du temps, déterminée par une fonction de décomposition de l'émotion.

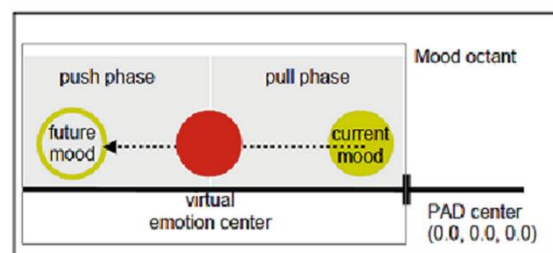


Figure 10 : La fonction de changement d'humeur push and pull

Nous concluons notre description de l'ALMA en expliquant le phénomène du changement d'humeur. L'humeur d'un agent peut changer au fil du temps. Pour modéliser ces changements d'humeur, nous utilisons les émotions comme facteur de changement d'humeur. L'humeur de l'agent peut changer à la suite de l'observation d'événements, d'actions ou d'objets qui suscitent des émotions qui s'atténuent au cours d'une période donnée. Les émotions actuellement actives sont représentées par un point correspondant dans l'espace PAD, après quoi la moyenne de tous ces points d'émotion est prise. Il en résulte un autre point dans l'espace PAD qui représente un centre "virtuel" des émotions actuelles de l'agent. Ce point est utilisé pour mettre à jour l'humeur actuelle en utilisant une fonction de changement d'humeur de type push and pull (voir Fig. 10). Si l'humeur actuelle se situe entre le centre virtuel des

émotions et le centre de l'espace PAD, l'humeur actuelle de l'agent est tirée vers le centre virtuel des émotions (phase d'attraction). En revanche, si l'humeur actuelle se situe au niveau ou au-delà du centre d'émotion virtuel, l'humeur actuelle de l'agent est poussée plus loin vers l'agent d'humeur actuel (phase de poussée). Ce processus de poussée peut à première vue sembler contre-intuitif, mais il représente le fait empirique que les expériences d'émotions qui sont cohérentes avec l'humeur actuelle d'un individu augmentent l'intensité de cette humeur. Plus l'intensité du centre d'émotion virtuel est élevée, plus son influence sur l'humeur actuelle de l'agent est forte. Notez que bien que l'humeur de l'agent soit influencée par ses émotions, l'humeur change plus lentement que les émotions elles-mêmes. Nous tenons à souligner que la Fig. 10 visualise à la fois les phases d'attraction et de répulsion. Cette illustration montre que l'humeur actuelle sera attirée vers le centre d'émotion virtuel lorsque l'humeur actuelle se trouve entre le centre de l'espace PAD et le centre d'émotion virtuel. Une fois que l'humeur actuelle est à la même position que le centre d'émotion virtuel, l'humeur actuelle sera repoussée du centre d'émotion virtuel plus loin dans l'espace PAD.

## Integration of 2APL and ALMA

Comme nous voulons intégrer les émotions dans 2APL, nous devons ajouter une base de données dans laquelle nous pouvons stocker les données d'évaluation. Ceci est fait dans ce qu'on appelle une base d'émotions. La base d'émotions stocke l'humeur actuelle de l'agent (sous la forme d'une valeur PAD) et l'émotion dominante (l'émotion avec l'intensité la plus élevée). La base d'émotions envoie l'ID de l'agent et une description des déclencheurs d'émotions (événements, actions ou objets) avec les balises d'évaluation correspondantes au composant AffectEngine. Ce composant ne fait pas partie du cadre ALMA ; il s'agit simplement d'une interface (classe/point d'entrée) que nous avons implémentée pour connecter 2APL à ALMA. Le composant AffectEngine convertit les étiquettes d'évaluation reçues en EEC, qui sont des entrées directes pour le composant AffectManager. Dans ce composant, qui fait partie intégrante du cadre ALMA [11], les EEC produisent des émotions spécifiques selon le modèle OCC, qui sont ensuite converties en valeurs PAD. Ces valeurs PAD modifient ensuite l'humeur de l'agent au moyen de la fonction de changement d'humeur push and pull (Fig. 10).

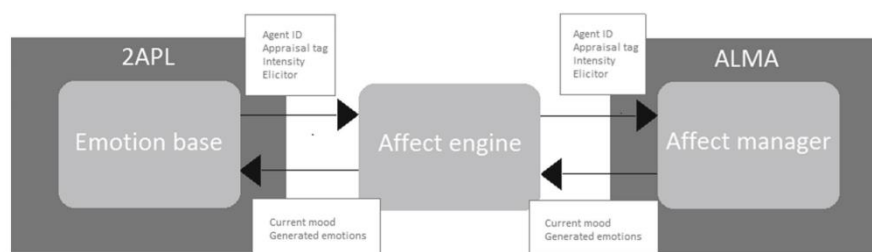


Figure 11 : La connexion entre 2APL et ALMA

La sortie du composant AffectManager est donc une nouvelle humeur (valeur PAD) ainsi que des intensités d'émotion mises à jour. Ce résultat est renvoyé au composant AffectEngine, qui détermine quelle émotion est générée par quel éliciteur. Ensuite, l'humeur et la correspondance entre l'éliciteur et l'émotion sont transmises à la base d'émotions de 2APL. La base des émotions met alors à jour l'humeur actuelle, l'émotion dominante et ce que l'on appelle les données d'adaptation (nous reviendrons plus tard sur les données d'adaptation,

mais il suffit pour l'instant de savoir qu'il s'agit d'une correspondance entre l'éliciteur et l'émotion).

## **Conclusion**

Dans ce chapitre, nous montrons comment un langage de programmation d'agents cognitifs peut être doté de moyens de programmer des émotions. En particulier, nous montrons comment le langage de programmation 2APL peut être augmenté afin qu'il puisse fonctionner avec le modèle d'émotion informatique ALMA pour traiter l'évaluation, la génération d'émotion/humeur et l'adaptation.

## Références

- 1- [https://www.researchgate.net/publication/265033357\\_FAtiMA\\_Modular\\_Towards\\_an\\_Agent\\_Architecture\\_with\\_a\\_Generic\\_Appraisal\\_Framework/references](https://www.researchgate.net/publication/265033357_FAtiMA_Modular_Towards_an_Agent_Architecture_with_a_Generic_Appraisal_Framework/references)
- 2- <https://www.biblio.univ-evry.fr/theses/2011/2011EVRY0023.pdf>
- 3- <https://docplayer.fr/11503025-Cor-e-un-modele-pour-la-simulation-d-agents-affectifs-fonde-sur-la-theorie-cor.html>
- 4- [https://www.researchgate.net/profile/Fabrice-Guillet/publication/266877718\\_Modelisation\\_des\\_agents\\_emotionnels/links/5717839f08ae2679a8c776ae/Modelisation-des-agents-emotionnels.pdf](https://www.researchgate.net/profile/Fabrice-Guillet/publication/266877718_Modelisation_des_agents_emotionnels/links/5717839f08ae2679a8c776ae/Modelisation-des-agents-emotionnels.pdf)
- 5- [https://www.researchgate.net/profile/Sylvie-Pesty/publication/228538673\\_Emotions\\_et\\_systemes\\_multi-agents\\_une\\_architecture\\_d%27agent\\_emotionnel/links/54294ddc0cf2e4ce940c9bcb/Emotions-et-systemes-multi-agents-une-architecture-dagent-emotionnel.pdf](https://www.researchgate.net/profile/Sylvie-Pesty/publication/228538673_Emotions_et_systemes_multi-agents_une_architecture_d%27agent_emotionnel/links/54294ddc0cf2e4ce940c9bcb/Emotions-et-systemes-multi-agents-une-architecture-dagent-emotionnel.pdf)
- 6- <https://sourceforge.net/projects/fatima-modular/files/>
- 7- <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.447.986&rep=rep1&type=pdf>
- 8- <https://commons.wikimedia.org/wiki/File:2APLplatform.png>
- 9- <http://mmi.tudelft.nl/~wouter/publications/pasman08b.pdf>