Modèle dynamique comportemental dans le cadre de la gestion de la prise de parole

Michaël Bernard

École Nationale d'Ingénieurs de Brest
Laboratoire en sciences et techniques de l'information, de la communication et de la connaissance,
Centre Européen de Réalité Virtuelle
Rapport de stage recherche, Master informatique spécialité Recherche en informatique 2012

Encadrant de recherche : Pierre Chevaillier

Modèle dynamique comportemental dans le cadre de la gestion de la prise de parole

Michaël Bernard

Plouzané, le 5 juin 2012

Résumé

Cette étude porte sur la conception d'un modèle dynamique de sélection de comportements dans un contexte d'interaction entre agents. Ce modèle continu abstrait se veut le plus simple et général possible pour pouvoir s'adapter à toutes sortes de comportements. Il se présente sous la forme d'un réseau proche d'un réseau neuronal et permet le calcul de valeur d'activation de comportement. Les calculs sont bâtis sur des entrées dépendant de l'intention propre à l'agent mais également du comportement des interlocuteurs. Ce modèle a été appliqué au comportement de prise de parole pour être implémenté par la suite au sein d'agents virtuels dans un collectif mixte d'humains réels et virtuels. Les comportements de parole et d'écoute ont été représentés et implémentés avec ce modèle sur les plateformes Scilab et Unity3D.

Mots-clés : conversation, agents virtuels, système dynamique, comportement, prise de parole, sélection d'action

Table des matières

Table des figures

1 Introduction

La réalité virtuelle a pour objectif de permettre à des utilisateurs de s'immerger dans des environnements, simulant un monde réel ou imaginaire, avec lesquels ils peuvent interagir de manière aussi naturelle que possible. Les environnements virtuels collaboratifs, tels que bon nombre d'environnements de réalité virtuelle pour l'apprentissage humain (EVAH), permettent à des utilisateurs d'interagir simultanément avec d'autres utilisateurs et, pour certains, également avec des humains virtuels.

Généralement, dans ce type d'application, les utilisateurs sont représentés dans l'environnement virtuel sous la forme d'un avatar humanoïde. La communication en langue naturelle entre utilisateurs et humains virtuels est un des modes possibles d'interaction. On peut par exemple avoir un échange entre un utilisateur et un agent afin de mettre en place une organisation de travail pour des tâches spécifiques ou pour la coordination de l'activité collaboratrice. Ce type de situation constitue le contexte de cette étude, à savoir la communication en langue naturelle au sein d'un collectif mixte d'humains réels et virtuels.

Ce travail s'intègre au projet CORVETTE (*COllaboRative Virtual Environment Technical Training and Experiment*) [1] qui a pour objectif le développement d'environnements d'apprentissage humain en s'appuyant sur la réalité virtuelle (Environnements de réalité Virtuelle pour l'Apprentissage Humain). Plusieurs axes sont visés par ce projet parmi lesquels on retrouve le travail collaboratif, l'humain virtuel, la communication entre un humain réel et un collaborateur virtuel, et l'évaluation des solutions innovantes proposées sur le plan de l'usage. Pour notre cas d'étude, nous rejoignons le point de la communication, avec, plus précisément, la gestion du tour de parole.

L'implémentation de la communication repose sur différents systèmes : la reconnaissance et la synthèse vocale, l'analyse et la synthèse des expressions faciales et de la gestuelle, le traitement de la langue naturelle (compréhension et génération d'énoncés), la gestion des tours de parole (*turn-taking*) et nous nous intéressons ici à ce dernier point.

Tour à tour, les agents adoptent un comportement de locuteur, attentif à son auditoire, et d'auditeur attentif au locuteur. L'agent doit être capable, en fonction du contexte, de sélectionner l'un ou l'autre de ces deux comportements. Cette décision intègre sa propre intention et les indices émis par les autres agents. La naturalité de l'interaction, aussi bien entre un utilisateur d'un humain virtuel et d'un agent qu'entre agents virtuels, repose sur une grande souplesse dans la transition entre les comportements de parole et d'écoute et sur la capacité des agents à les adapter dynamiquement [ter Maat et al., 2010], [Yuasa and Mukawa, 2011].

L'étude porte sur la modélisation de l'activation des comportements de parole et d'écoute par un agent conversationnel. L'objectif est d'assurer des transitions souples entre les comportements afin de simuler la naturalité des changements de locuteurs.

Notre objectif est de définir les bases d'un modèle dynamique de contrôle de la sélection d'actions pour des agents en interaction avec un utilisateur dans un environnement virtuel. Ce modèle se veut générique, adaptable à différents comportements et prises de décisions, et continu. Il repose sur un système proche d'un réseau neuronal et intègre des entrées internes et externes. Nous avons particulièrement étudié notre modèle dans le cadre de la gestion de prise de parole, d'abord avec un comportement de parole puis par la cohabitation des comportements de parole et d'écoute.

Dans un premier temps, nous présentons quelques travaux existants ainsi que des

1. `http://corvette.irisa.fr/`

définitions qui seront utiles à la compréhension de cet écrit, puis nous allons détailler notre modèle comportemental et voir comment nous l'avons adapté aux comportements conversationnels que sont la parole et l'écoute. Nous avons également implémenté le modèle afin de l'observer au sein d'un agent et nous allons présenter les travaux que nous avons réalisés avant de discuter de ce modèle et des perspectives associées.

2 Travaux existants

Avant de commencer à détailler notre étude, nous nous arrêtons sur quelques points de définition et d'explication sur les propriétés des conversations entre humains. Ensuite nous présenterons deux modèles existants de prise de parole, déterministe et non déterministe, tous les deux basés sur des modèles de type états-transitions.

2.1 Conversations humaines

Concernant nos travaux, nous allons nous appuyer sur les définitions suivantes pour la suite de cette étude. Plusieurs termes sont utilisés et ont un sens particulier qu'il est important de bien saisir.

2.1.1 Définitions

Tout d'abord, on parle de « tour de parole » lorsque l'on est en position de locuteur. Le locuteur est celui qui exprime une idée (un contenu conversationnel) à travers le discours. Il est en position de parole. L'auditeur sera celui qui, au contraire, sera en position d'écoute. On parlera d'interlocuteur pour signifier que la personne est susceptible de prendre l'une ou l'autre des positions au cours de la conversation.

On passe de locuteur à auditeur suite à un changement de tour de parole (*turn-taking*). Le locuteur devient auditeur (sauf cas particuliers de chevauchement de parole, non étudiés ici), et un auditeur devient locuteur.

Lors d'une conversation, l'objectif est d'exprimer son désir conversationnel à travers le contenu du discours, la présentation du discours, les gestes mais aussi par l'écoute. Ces mécanismes sont multiples, et parmi eux on retrouve l'échange d'indices de deux types pour les conversations : invitations et signaux.

Le locuteur produira des indices différents en fonction du contenu, du moment de la conversation ainsi que le moment de son tour de parole. Ainsi, lorsque celui-ci s'exprime, il pourra inviter l'auditeur à produire des signaux de retour (*backchannels inviting cues*, [Gravano and Hirschberg, 2009]). Lorsque le locuteur termine son tour de parole, il produira des invitations de prise de parole (*turn-taking inviting cues*, [Gravano and Hirschberg, 2011]).

En tant qu'auditeur, le rôle n'est pas passif. En effet, il faut exprimer pour le locuteur son ressenti de perception, compréhension, accord avec le discours et cela passe également par les indices. Ils sont de même type mais leur expression est différente. Il s'agit là de signaux de retour (*backchannels*, [Bevacqua, 2009]). Ces signaux n'impliquent pas un changement dans le tour de parole.

De plus, l'auditeur produit des signaux de retour à son gré, et/ou suivant les invitations du locuteur, mais également des signaux de prise de parole. Ces signaux indiquent au locuteur que son auditeur a un désir plus ou moins important de prise de parole.

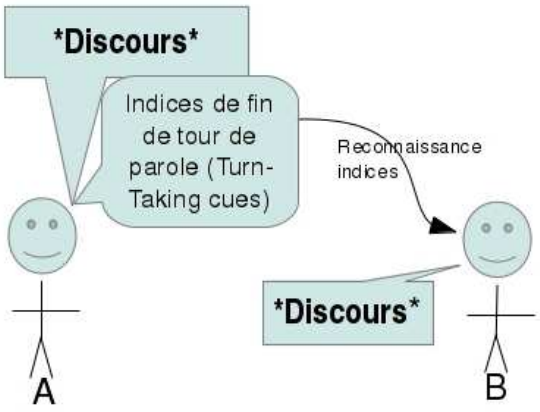


FIGURE 1 – Prise de parole de B (auditeur) après A (locuteur) en suivant des indices de fin de tour verbaux.

À présent que les définitions ont été explicitées, nous allons nous arrêter sur les conversations humaines afin de mieux comprendre leur fonctionnement et de pouvoir les modéliser par la suite.

2.1.2 Modélisations des conversations humaines

Ce qui nous intéresse ici ce sont les changements de prise de parole (*turn-taking*) effectués entre les différents interlocuteurs. Afin de voir ces changements s'effectuer au mieux, il existe entre humains des codes implicites régissant le passage de tour de parole qui mettent en jeu des signaux verbaux et non verbaux. Ainsi, dans des relations polies, l'auditeur ou auditoire sait à quels moments il peut intervenir et à quels moments il ne peut pas et donc ne coupe pas la parole de son locuteur (voir figure 1). Un modèle de transition de parole a été présenté en 1974 par [Sacks et al., 1974], nommé SSJ, et cité par [Gravano and Hirschberg, 2011]. Il est construit sur des observations et des faits apparents lors de discussions. Il a permis d'introduire la notion de TRP (*Transition-Relevance Place*), c'est-à-dire des moments plus ou moins pertinents de changement de locuteur.

Cette notion n'a pas été formellement définie mais a permis à d'autres travaux d'en découler. En s'appuyant dessus, on peut ensuite chercher à identifier des signaux de fin de tour. Ces derniers sont multiples mais sont identifiables lors de discussions. Le ton qui change en fin de phrase, l'achèvement d'une proposition ou encore l'arrêt de mouvements gestuels accompagnant la parole peuvent indiquer que le locuteur a fini de parler, ou est sur le point de le faire, et que l'on se trouve à ce moment précis en TRP.

Durant un TRP, plusieurs choses sont possibles : si la personne qui avait la parole a sélectionné quelqu'un pour le prochain tour, cette personne est censée intervenir ; sinon, n'importe qui peut prendre la parole. Enfin, si personne ne prend la parole, le locuteur peut reprendre la parole et commencer à parler de nouveau.

Souvent, les indices marquants pour changer le tour de parole sont portés par l'intonation et la fin d'une proposition. Certaines intonations peuvent aussi être significatives de l'envie de garder le tour de parole. De plus, l'analyse d'enregistrements vidéos de conversations entre trois personnes a montré que le regard jouait également un rôle dans le changement de locuteur [Yuasa et al., 2009]. Ces observations ont mis en évidence les trois règles suivantes dans la réalisation de tours de parole :

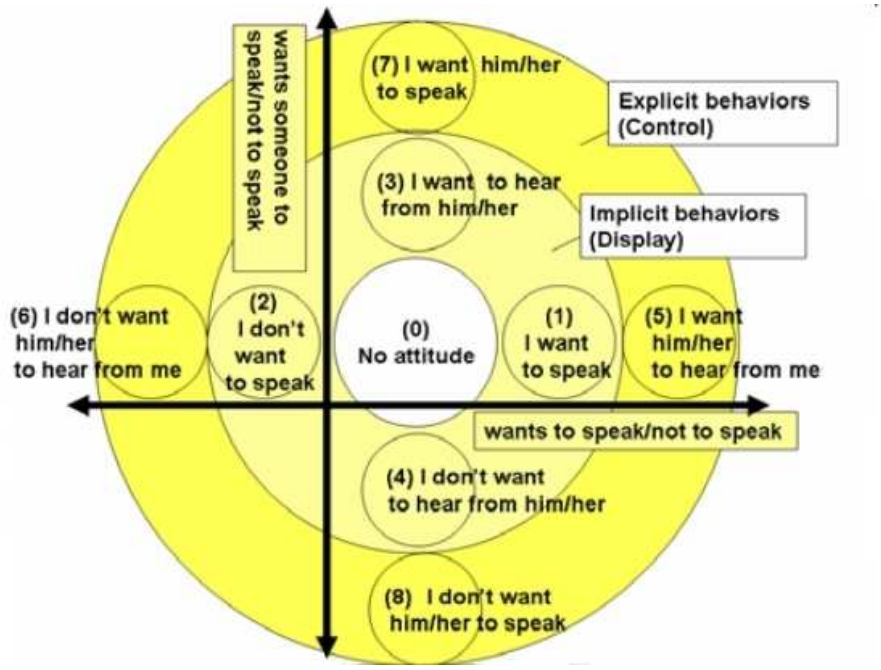


FIGURE 2 – Modèle d'attitude de parole extrait de [Yuasa and Mukawa, 2011].

1. la personne regardée par le locuteur venant de finir son tour prend le tour de parole ;
2. une personne non regardée par le locuteur venant de finir son tour, prend la parole ;
3. un des auditeurs prend la parole même si le locuteur ne regarde personne.

Après l'analyse des conversations, les auteurs ont pu montrer que les trois règles s'appliquaient à des taux différents. La première est respectée dans 65% des cas, et apparaît comme majeure, tandis que la seconde n'est vérifiée que dans 26% et enfin la dernière dans 9% des cas. Cette étude montre bien que suivant le comportement physique du locuteur, l'auditoire ne réagit pas de la même façon. Ensuite, en analysant plus méticuleusement les comportements, il s'est avéré que le regard n'était pas le seul indice de transition du tour de parole, et que la gestuelle était elle aussi importante [Yuasa and Mukawa, 2011].

Ces auteurs ont réalisé un avatar gérant le tour de parole en s'exprimant à partir de leur propre modèle d'attitude. Ce modèle permet de catégoriser des comportements utilisés dans le tour de parole. Ils distinguent 9 cas différents pour le tour de parole qu'ils organisent sur deux axes : je veux parler/ne pas parler ou je veux écouter untel ou non (figure 2).

De plus, ils ont fait la distinction entre des comportements implicites (on s'attend à ce que les autres les remarquent et les comprennent) et explicites (permettent de contrôler de manière intentionnelle et directe les comportements de parole des autres). Sur la figure 2, le premier cercle en partant du centre représente les comportements implicites tandis que le cercle le plus éloigné du centre représente les comportements explicites. On trouve donc 2 comportements pour chacun des 2 axes (un positif, un négatif) et pour chacun des types de comportements. Ils prennent aussi en compte le comportement de « non-attitude » ce qui apporte le 9^{e} cas pour la gestion du tour de parole.

Toujours dans ce contexte, d'autres auteurs proposent non plus de s'appuyer sur la notion de TRP, mais sur celle d'IPU (*Inter-Pausal Unit*) qui représente une séquence de mots entourée d'un silence d'une durée minimale de 50 ms [Gravano and Hirschberg, 2011]. À partir de là, ils ont mis en place une étude afin de déterminer des évènements mesurables

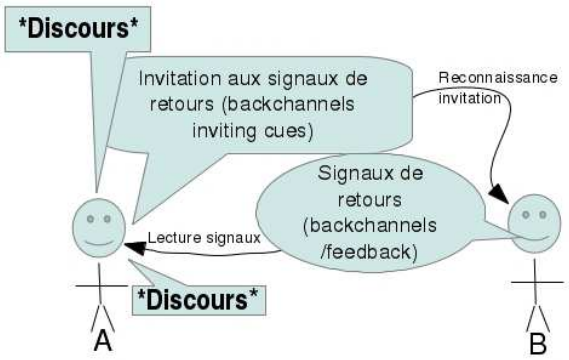


FIGURE 3 – Retour de B (auditeur) après invitations A (locuteur), puis reprise du discours par A.

(indices) qui rendent possible le changement de locuteur. Parmi ces évènements, on trouve :

- un changement d'intonation à la fin d'une IPU (intonation montante ou descendante) ;
- des IPU plus courtes (les séquences de mots vont devenir de plus en plus courtes, montrant que le locuteur a peut être tout dit sur le sujet) ;
- une intensité de volume de voix plus faible ;
- une vibration vocale plus grande que lors du discours ;
- une durée de silence plus longue entre les phrases du locuteur.

Ces indices permettent d'annoncer une possibilité de changement de tour de parole, et plus ces derniers sont nombreux, plus ils vont inciter une personne de l'auditoire à prendre la parole.

À présent, voyons les invitations au retour (*backchannels-inviting cues*) qui invitent l'auditoire à réagir au discours du locuteur et d'afficher son accord, désaccord, ou incompréhension (*backchannels* ou *feedbacks*).

Ces retours permettent au locuteur d'adapter son discours (niveau de langue, répétition d'information pour faciliter la compréhension, etc.). Mais pour cela, le locuteur doit lui-même transmettre à l'auditoire des indices qui indiquent sa volonté de recevoir un retour ou qui permettent juste à l'auditoire d'en donner (voir figure 3).

Classiquement, l'auditoire peut répondre aux invitations de retour par de courtes onomatopées ("ok", "mmm", etc.) et/ou par de la gestuelle (hochement de tête, regard interrogatif, etc.).

Les indices ne se produisent pas n'importe quand dans la conversation. Ils se situent très majoritairement proche ou durant un TRP, ont une intonation non finale (c'est-à-dire qui n'indiquent pas une fin de tour de parole) et sont suivis de la continuité du discours du locuteur.

Gravano et collaborateurs ont mis en évidence l'existence de plusieurs indices qui invitent à la production de retours [Gravano and Hirschberg, 2011] :

- une intonation montante à la fin d'une IPU ("n'est-ce pas ?")
- une intensité de volume de voix plus forte
- des suites de mots significatives (déterminant + nom, adjectif + nom, ou nom + ponctuation)
- une vibration vocale plus faible que lors du discours
- une durée de silence plus longue entre les phrases du locuteur

Et tout comme pour les indices de changement de parole, plus les indices présents sont nombreux, et plus l'auditoire sera incité à fournir des retours.

Il faut noter tout de même que dans les deux cas (prise de parole et retours), même si les invitations sont présentes, elles n'obligent en rien l'auditoire à réagir.

Ces études sur les conversations humaines ont permis à d'autres auteurs de proposer des modèles informatiques pour la gestion du tour de parole qui reposent sur les signaux verbaux et ne prennent pas en compte la gestuelle ni les postures des interlocuteurs. Ces modèles nous intéressent justement sur ce point et nous allons les détailler maintenant.

2.2 Modèles de tour de parole pour un agent conversationnel

Deux modèles ont attiré notre attention, celui de Kronlid et celui de Raux et Eskenazi. Ils sont différents l'un de l'autre, ayant chacun des caractéristiques particulières intéressantes, et c'est ce que nous allons voir par la suite en débutant par le modèle de Kronlid.

2.2.1 Modèle de prise de parole de [Kronlid, 2008]

Le modèle présenté est un gestionnaire de prise de parole interne à l'agent. Bâti sur le modèle SSJ [Sacks et al., 1974], et sur les *statecharts* de Harel [Harel, 1987], l'agent va recevoir des évènements qui lui indiquent ce qu'il se passe (ce participant a commencé à parler, à se taire...), et adapter ses états en conséquence. Ce modèle a d'abord été construit pour des conversations en face-à-face puis a été étendu pour des conversations multi-parties. Ce modèle est organisé en trois différents *statecharts* :

1. *outside statechart* ;
2. *inside statechart* ;
3. *TRP statechart*.

Le premier représente le monde extérieur du point de vue interne à l'agent. Il permet de connaître l'état des partenaires de discussion (en train de parler ou silencieux), donc deux états. L'auteur présente cette version simple du *statechart* et une version plus compliquée dans laquelle il ajoute des prédictions sur l'état des autres (prédictions d'arrêt ou de prise de parole).

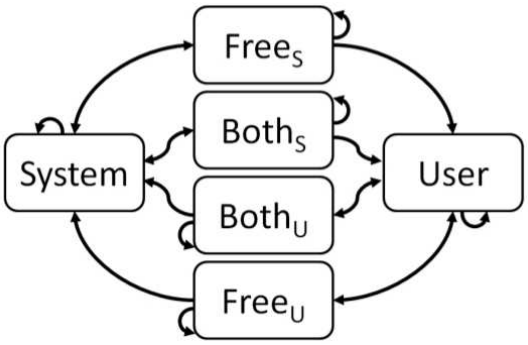
Le second signale quand l'agent parle au même instant qu'un interlocuteur. On a dans ce *statechart* deux états (l'agent parle ou l'agent se tait), et dans le premier état, deux états internes représentent la présence de conflits (un autre interlocuteur parle en même temps).

Enfin, le dernier *statechart* représente les TRP vus dans la partie précédente. Celui-ci dispose également de deux états internes (avoir ou non un TRP).

Cette architecture est déterministe, chaque état est atteint par un évènement spécifique, et est également extensible aux conversations multi-parties. Ici, il s'agit d'un gestionnaire de prise de parole où les tours de parole ont été modélisés. Tout ce qui se rapporte à la gestuelle, au regard, mouvements de tête et d'épaule ainsi que la production ou interprétation de retours ou contenus n'est pas pris en compte.

2.2.2 Modèle de prise de parole de [Raux and Eskenazi, 2009]

L'autre modèle s'intéressant à la prise de parole est celui proposé par [Raux and Eskenazi, 2009]. Comme le précédent, c'est un modèle basé états-transitions,



(a) Modèle de la dynamique du tour de parole

State \ Action	K	R	W	G
$SYSTEM$	0	C_S	-	-
$BOTH_S$	$C_O(\tau)$	0	-	-
$BOTH_U$	$C_O(\tau)$	0	-	-
$USER$	-	-	0	C_U
$FREE_U$	-	-	$C_G(\tau)$	0
$FREE_S$	-	-	$C_G(\tau)$	0

(b) Matrice de coût des actions dans chaque état. Légende - : action non disponible

FIGURE 4 – Modèle de Raux et Eskenazi, extrait de [Raux and Eskenazi, 2009].

mais celui-ci est non déterministe, c'est-à-dire qu'on passera d'un état à un autre non plus suivant des évènements fixés par avance, mais selon des coûts de passage. Reprochant à certains modèles leurs trop grande complexité de développement, les auteurs s'appuient sur des diagrammes à états finis. S'intégrant dans une conversation bipartie, Raux et Eskenazi proposent un modèle à six états (voir figure 4(a)).

Ces états sont inspirés des travaux de [Jaffe and Feldstein, 1970] sur le dialogue, et sont définis comme étant les états d'intentions et d'obligations des interlocuteurs plutôt que des états de silence ou temps de parole.

Ainsi $System$ et $User$ représentent l'instant où, respectivement, l'agent ou l'utilisateur souhaite prendre la parole ou a l'obligation de la prendre (après une question par exemple).

Les états $Both_U$ et $Both_S$ indiquent lorsque les deux protagonistes ont la parole en même temps. Le $_U$ ou le $_S$ indique l'état précédent dans lequel on se trouvait, respectivement $User$ et $System$, et les auteurs ont volontairement admis qu'il était rare, et donc négligeable, que les deux interlocuteurs commencent à parler à l'exact même instant.

De plus, il faut noter que toutes les transitions ne sont pas bidirectionnelles (voir figure 4(a)).

Pour passer d'un état à un autre quatre actions sont possibles :

(r) *release*, laisser la parole ;
(g) *grab*, prendre la parole ;
(w) *wait*, attendre la fin d'un tour de parole ;
(k) *keep*, garder la parole ;

Ces actions ne sont pas toutes activables suivant l'état dans lequel se trouve la machine (toujours deux sur quatre). En effet, dans l'état $System$, l'agent ne peut pas « prendre » (g) puisqu'il dispose déjà du tour de parole par exemple. Pour choisir une action à effectuer, une structure de coût est proposée, dans laquelle il est fait l'hypothèse que les participants tentent de minimiser les silences et les dialogues simultanés (chevauchements).

Les auteurs ont proposé trois règles pour mettre en place une matrice de coût.

1. Le coût d'une action qui résout un silence ou un chevauchement vaut zéro.
2. Le coût d'une action qui crée un silence ou chevauchement non souhaité est égal à un paramètre constant (potentiellement différent pour chaque paire action/état).
3. Le coût d'une action qui maintient un silence ou chevauchement est soit une constante ou une fonction d'incrémentation du temps passé dans cet état.

Il en découle une matrice de coût (voir figure 4(b)) dans laquelle :
- C_S est le coût d'interruption d'un message système avant sa fin alors que l'utilisateur n'a pas demandé la parole ;
- $C_O(\tau)$ est le coût de maintien d'un état de chevauchement qui est déjà long de τ ms ;
- C_U est le coût de demande de parole alors que l'utilisateur l'a déjà ;
- $C_G(\tau)$ est le coût de maintien d'un état de silence qui est déjà long de τ ms.

La décision optimale sera celle qui minimise le coût attendu d'une action A donné par :

$$\mathcal{C}(A) = \sum_{S \in states} P(s = S|O) \times C(A, S) \quad (1)$$

où *states* est l'ensemble des états, O les caractéristiques observables du monde, $P(s = S|O)$ la probabilité que l'utilisateur demande la parole et $C(A, S)$ est le coût de l'action A dans l'état S suivant la matrice de coût.

3 Modèle comportemental

Ces modèles existants nous ont intéressés pour leur spécificités d'états-transitions ainsi que le non déterminisme du modèle de Raux et Eskenazi. Cependant, nous avons développé notre modèle en partant sur une nouvelle base de travail qui n'avait pas été étudiée par ces deux précédents modèles. De plus, nous l'avons conçu afin que celui-ci puisse s'intégrer aisément au sein d'architectures existantes (YTTM [Thórisson, 2002], SAIBA [Kopp et al., 2006]) notamment au sein de l'étape de planification du comportement (*behavior planning*) de SAIBA sur laquelle nous reviendrons par la suite. Dans cette partie, nous commençons par détailler les principes sur lesquels reposent ce modèle avant de voir de plus près le modèle en lui-même.

3.1 Principes

Notre modèle repose sur différents principes tels ceux étudiés dans la littérature existante, mais également des hypothèses que nous avons faites afin de proposer un modèle qui correspond aux observations que nous avons réalisé lors de notre études de la gestion du tour de parole dans les conversations humaines. Nous allons présenter ces hypothèses et expliquer pourquoi ces choix ont été faits.

3.1.1 Hypothèses

La vision de départ sur laquelle repose le développement de ce modèle est que les conversations humaines, et plus précisément les comportements peuvent être représentés comme des systèmes dynamiques. Le tour de parole passe d'un interlocuteur à un autre ni selon un plan préétabli ni au hasard mais plutôt de façon dynamique : ce qui se passe, finalement, est déterminé dynamiquement au cours de l'interaction.

Nous sommes partis du postulat qu'un comportement se crée de manière progressive dans le temps et fait suite à des évènements internes et externes et qui influent sur l'intention d'action.

Une autre hypothèse qui a été faite concerne le déclenchement d'un comportement. Nous avons souhaité que les comportements soient représentés par des valeurs continues, et par conséquent, un comportement sera déclenché ou non suivant une valeur de seuil.

Et suite aux écrits de Frank et collaborateurs ([Frank et al., 2009]), nous avons considéré que le seuil de déclenchement d'un comportement n'est pas le même que le seuil d'arrêt. Nous avons essayé par la suite de prendre en compte cette notion d'hystérésis dans le développement de notre modèle.

3.1.2 Type de modèle et champ de la modélisation

Le modèle présenté par la suite répondra à diverses caractéristiques, jugées utiles à son bon fonctionnement.

Pour commencer, il a tout d'abord été construit pour être le plus abstrait possible, présentant une organisation permettant de décrire ou de pouvoir implémenter des comportements de différentes natures. Dans notre cas, nous l'avons adapté à des comportements antagonistes de parole et d'écoute, comportements a priori binaires (on se parle ou on se tait - on écoute ou on ne fait pas attention) mais notre modèle est un modèle continu.

En effet, un comportement pourra être défini comme activé ou non, mais pour autant, son activation dépend de plusieurs variables qui influent globalement sur le comportement. C'est pourquoi il nous est apparu justifié de mettre en place un modèle continu.

Nous avons exclu du champ de la modélisation la gestion de l'intention de l'agent à adopter un comportement, c'est-à-dire, ce qui fait que le locuteur a quelque chose à dire ou que l'auditeur est intéressé par ce qui se dit. Pour les besoins de notre étude les intentions, de parler ou d'écouter, constituent des forçages du modèle.

Enfin, notre modèle ne concerne pas non plus la gestion de contenus. Nous sommes restés sur le dynamisme même des comportements. Dans notre cas, aucun traitement n'est effectué en fonction des énoncés, et notre modèle s'arrête avant la phase de réalisation de comportement (*behavior realisation*).

En résumé, nous nous intéressons ici à la simulation du mécanisme de sélection des comportements et non à celle de sa réalisation.

Maintenant que nous avons approché les principes du modèle, nous allons présenter l'organisation de ce dernier dans le cadre de comportements plus spécifiques afin de l'étudier et pouvoir par la suite proposer une implémentation de ces comportements.

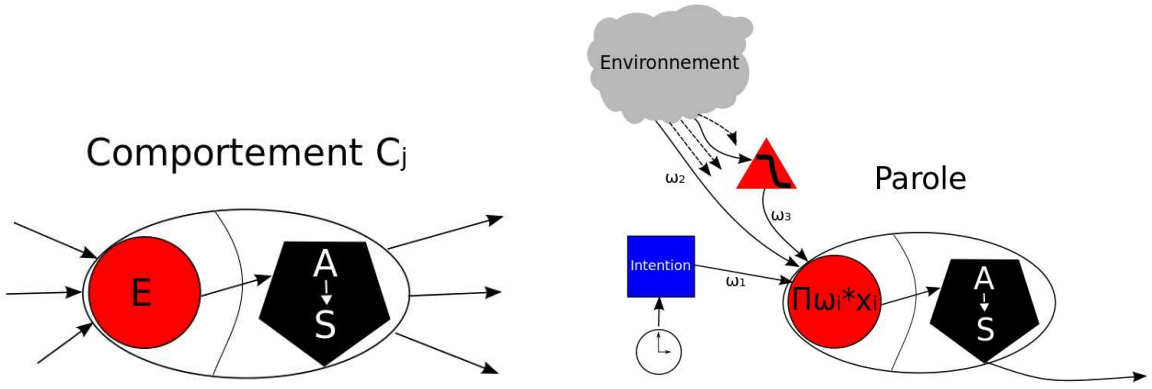
3.2 Comportement de parole

Nous allons commencer ici par étudier le modèle dans un comportement de parole en présentant tout d'abord les hypothèses faites en fonction des objectifs que nous souhaitions atteindre pour ce cas, avant de détailler le modèle d'activation de ce comportement, et enfin les productions inhérentes au tour de parole.

3.2.1 Hypothèses - objectifs

Pour le comportement de parole, nous souhaitions que le modèle soit le plus robuste et générique possible. On a eu recours à des simplifications de différents types.

Pour commencer, le modèle ne prend pas en compte le contenu des conversations. Tout est bâti sur la dynamique de l'interaction, et le comportement produit. Par conséquent, tous les mécanismes du tour de parole liés au contenu et à la prosodie ne sont pas utilisés dans le modèle comportemental. On a fait l'hypothèse que ces mécanismes sont gérés ailleurs, dans un autre modèle dédié.



(a) Unité de base du réseau (b) Schéma du réseau pour le comportement parole

FIGURE 5 – Schémas du modèle comportemental.

En ce qui concerne la production du comportement de parole, nous avons fait l'hypothèse que son activation est motivée par une intention propre à l'agent (l'intention de parler) mais également d'indices extérieurs tels que le comportement ou actions des autres interlocuteurs.

Dans cette première version du modèle, nous faisons l'hypothèse que l'activation du comportement de parole d'un agent est conditionnée par son intention de parler et par l'exécution du comportement de parole par un autre agent.

3.2.2 Modèle d'activation du comportement de parole

L'architecture du modèle se présente sous la forme d'un ensemble de cellules interconnectées que nous appelons ici neurones car elles ont des propriétés similaires aux neurones formels des réseaux de neurones. Chaque neurone simule l'activation potentielle d'un comportement.

Dans la représentation graphique de notre modèle, les arcs entre les neurones représentent soit une activation soit une désactivation. La notation par défaut est celle d'un lien d'activation ; un triangle indique une désactivation (figure 5(a)), le cercle est le point d'entrée du neurone et l'hexagone le point de sortie. On disposera d'entrées qui seront prises en compte dans un premier calcul (E) avant de passer dans une fonction d'activation (A) et de produire en sortie une valeur d'activation (S) de comportement ainsi que d'autres sorties suivant les besoins.

La fonction d'entrée E utilisée pour le comportement de parole a été choisie comme étant un produit pondéré des entrées (équation 2). Pour le modèle, il faut prendre en entrée l'intention, et le comportement des autres pour le moment (équation 3). S'agissant d'un réseau bouclé, les valeurs en entrée d'un comportement à l'instant t sont calculés à partir des activations des autres comportements à $t - \Delta t$. De plus, l'entrée ne correspond pas exactement aux comportements, mais aux comportements qui sont préalablement passés dans une fonction de désactivation.

$$\prod_i w_i \times x_i \tag{2}$$

$$\begin{aligned} E : \mathbb{R}^+ &\to [0;1] \\ t &\mapsto w_1.\iota_1(t) \times w_2.\mathcal{F}_{D1}(\mathcal{F}_{A2}(E_2(t-\Delta t))) \end{aligned} \tag{3}$$

$\iota_1(t)$ correspond à la valeur d'intention de parole citée précédemment. $\mathcal{F}_{D1}$ étant la fonction de désactivation du premier interlocuteur à laquelle on passe en entrée l'activation du comportement du second interlocuteur en présence (fonction $\mathcal{E}_2$). Les deux fonctions qu'il reste à décrire sont les fonctions d'activation et désactivation. L'une va servir à produire une valeur d'activation de comportement, quand l'autre permet la désactivation des comportements des autres pour le calcul de la valeur d'activation.

Ce sont toutes deux des fonctions sigmoïdes définies de la façon suivante :

$$\begin{aligned} \mathcal{F} : \mathbb{R} &\to [0;1] \\ x &\mapsto \frac{1}{1+\exp^{-\alpha\times(x-\eta)}} \end{aligned} \tag{4}$$

α est positif pour une fonction d'activation et négatif pour une fonction de désactivation. Sa valeur absolue contrôle la sensibilité de l'activation d'un comportement à partir de l'activation (potentielle) d'autres comportements. Le paramètre η contrôle la valeur du seuil de réalisation du comportement ($\mathcal{F}(x) = 0.5$ pour $x = \eta$).

La valeur de sortie de la fonction d'entrée E sera ensuite passée dans la fonction $\mathcal{F}_A$ pour produire la valeur d'activation du comportement.

On a présenté ici le fonctionnement interne du modèle, mais il faut à présent s'attarder à la valeur en sortie de la fonction d'activation qui va permettre, entre autre, la production d'indices.

3.2.3 Production des indices liés au tour de parole

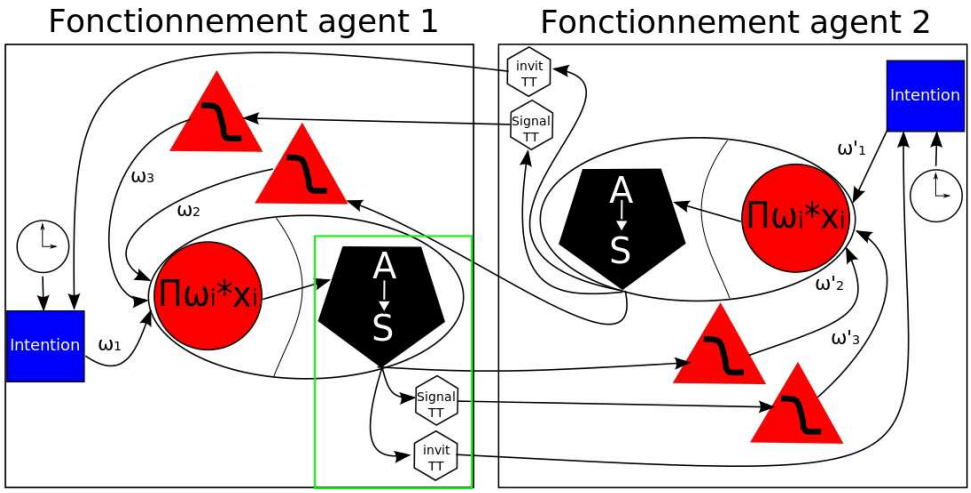
À la sortie de la fonction d'activation, nous avons une valeur, que nous utilisons comme variable de contrôle de la production des indices liés à la parole, c'est-à-dire les invitations à la prise de parole, et également aux signaux de prise de parole. En effet, pour simplifier les choses, nous avons décidé dans un premier temps de ne prendre en compte que le comportement de parole, donc le comportement d'auditeur ne sera pas modélisé, et la production des signaux de prise de parole correspondra à une augmentation de la valeur d'activation du comportement de parole.

Les invitations et signaux de la prise de parole dans les conversations humaines sont de plusieurs formes (voir section 2.1) et nous avons choisi de les représenter ici par une valeur comprise entre 0 et 1, calculée en fonction de la valeur d'activation. Pour cela, nous avons émis l'hypothèse que lorsque la valeur d'activation du comportement est en croissance, les signaux de prise de parole se faisaient plus forts, et inversement, lorsque la valeur d'activation du comportement est en décroissance, les invitations à la prise de parole étaient plus importants. Donc nous avons utilisé la dérivée instantanée de la fonction d'activation ($\dot{\mathcal{F}}_A$). Ce calcul de signaux et invitations se fait après calcul de la valeur d'activation du comportement (figure 6(a)).

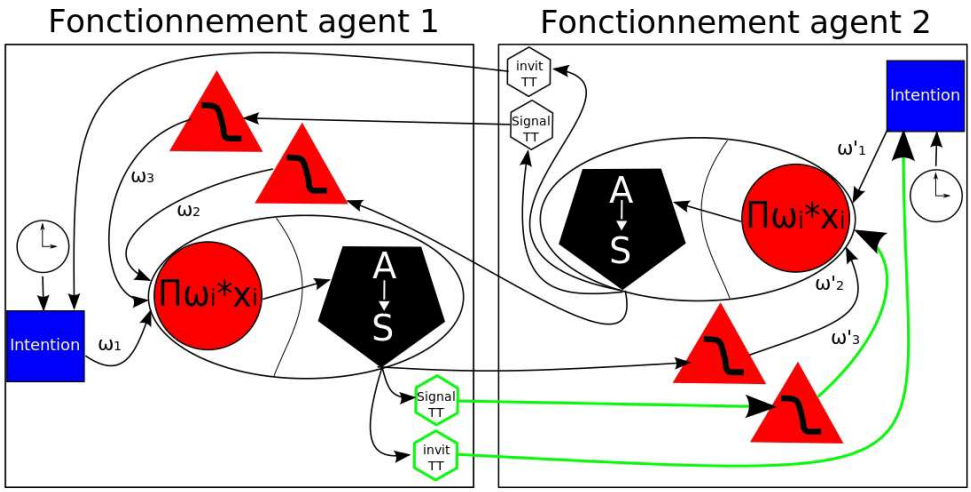
Pour la production de valeur de signal, on prendra l'opposé de cette fonction afin d'obtenir des valeurs supérieures à zéro.

En ce qui concerne l'interprétation de ces indices, rappelons leur utilité. Les invitations permettent de catalyser le comportement de parole, quand, au contraire, les signaux auront un effet inhibiteur. Ces indices vont arriver en entrée du modèle, dans la prise en compte du calcul de la valeur d'activation du comportement (figure 6(b)).

Les invitations produisent une augmentation de la valeur du poids de prise en compte de l'intention. En effet, comme les invitations catalysent le comportement, une prise en



(a) Production



(b) Interprétation

FIGURE 6 – Production et interprétation des invitations et signaux de prise de parole.

compte importante de l'intention de parler va permettre une valeur d'activation de comportement plus forte plus rapidement. On additionne la valeur d'invitation au poids de l'intention de parole.

Pour les signaux, la situation est légèrement différente. Comme ces derniers inhibent le comportement, on doit, avant de les prendre en compte, les faire traverser la même fonction de désactivation utilisée pour la désactivation liée aux comportements des autres (équation 4). La valeur calculée par la fonction de désactivation sera ensuite passée en entrée de la fonction d'entrée E, (produit pondéré, équation 3)

Il faut noter que pour ces indices, on ne prend pas forcément le dernier produit. En effet, cela poserait des problèmes si on mettait en place ce modèle interne en concurrence à lui-même avec l'exacte même intention en entrée. On obtiendrait des comportements de miroirs qui ne sauraient comment s'équilibrer. C'est pourquoi nous avons proposé que les calculs se basent sur n derniers indices. Le n est à définir suivant les cas, et peut représenter la capacité du modèle à garder en mémoire les derniers indices reçus. Les calculs se font donc en prenant l'un de ces n derniers indices aléatoirement.

Ce comportement de parole n'est pas le seul existant dans une conversation, et est plutôt limité s'il n'y a pas en réponse un comportement d'écoute. C'est pourquoi nous avons souhaité le mettre en place afin d'observer les changements dynamiques entre ces deux comportements et observer si le modèle répondait à une dynamique du tour de parole.

3.3 Comportements de parole et d'écoute

Nous venons de décrire le modèle appliqué à un comportement de parole et ce qu'il en était pour les indices conversationnels. À présent, nous souhaitons étendre le modèle en intégrant le comportement d'écoute. Pour cela, il nous a fallu poser des hypothèses liées à la dynamique du tour de parole afin de proposer une modélisation de ce comportement. Nous allons commencer par présenter ces hypothèses avant d'observer quel est le couplage entre les comportements de locuteur et d'auditeur et enfin de détailler la dynamique de la sélection d'action de ces deux comportements.

3.3.1 Hypothèses

L'écoute est un comportement actif. Il se traduit par l'émission d'indices permettant au locuteur d'adapter son discours en fonction, par exemple, de la compréhension transmise par l'auditeur. Mais un comportement d'écoute ne peut être présent seul, il n'existe que par l'existence en face d'un comportement de parole (on n'écoute pas une personne qui ne parle pas).

Nous avons posé l'hypothèse que l'agent peut parler ou écouter, mais pas les deux à la fois. Il peut ne pas être en train de parler ni d'écouter (personne ne parle).

Dans un premier temps, nous avons considéré que l'agent parle uniquement s'il a quelque chose à dire, et que personne ne parle. Il arrête de parler quand l'un de ces deux critères est manquant, ou lorsque plus personne n'écoute.

Inversement, l'agent écoute lorsque quelqu'un parle et lorsqu'il n'a pas envie de parler. Si l'envie de parler se fait sentir, et qu'il se trouve en état d'écoute, alors il désactive son comportement d'écoute.

Le comportement d'écoute est déclenché par un comportement de parole actif chez un autre agent. On prend pour simplifier la valeur d'activation du comportement de parole

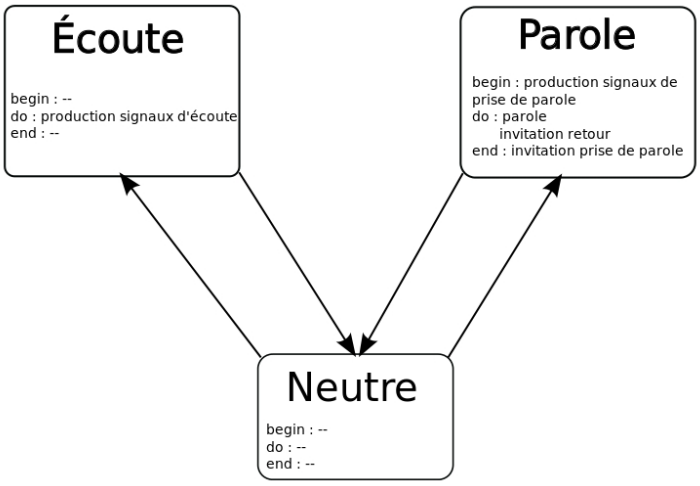


FIGURE 7 – Activités de l'agent en fonction de son comportement.

pour calculer la valeur d'activation du comportement d'écoute.

Lors d'une écoute, l'agent ne produit pas des signaux uniquement lorsqu'il est invité à le faire, donc la production de signaux doit être régulière mais également réactive vis-à-vis des invitations.

Afin de pouvoir faire cohabiter les comportements entre eux, nous avons besoin de déterminer l'état des entités en présence. Par conséquent, on peut déterminer l'état actuel d'un système mis en confrontation (c'est-à-dire que l'on sait si un système est à l'écoute ou en parole par exemple).

Ces hypothèses nous ont servi de base à la construction de notre modèle comportemental adapté à l'écoute attentive. Nous allons à présent détailler comment ce dernier a pu être mis en place et cohabiter avec un comportement de parole.

3.3.2 Couplage des comportements du locuteur et de l'interlocuteur

Pour coupler les deux comportements et répondre aux hypothèses, nous avons tout d'abord imaginé 3 états qui sont « parole », « écoute » et un état « neutre ». Ce dernier correspond au moment où l'agent n'est ni en écoute, ni en parole (figure 7).

À chaque état correspond l'activation d'un comportement et de production de signaux. Ainsi, lors de la parole, on aura la production des signaux de prise de parole à l'entrée de cet état, la production des invitations aux retours, et quand on quitte l'état, l'agent émet des invitations à la prise de parole.

Concernant l'état d'écoute, l'agent devra produire de manière régulière des signaux d'écoute. Pour cela, la valeur de signal se verra être à 0 en dehors de l'écoute, comprise entre 0.5 et 1 pendant l'écoute, suivant le comportement de parole et les invitations aux retours. La valeur d'activation du comportement d'écoute sera calculée par le même type de réseau neuronal que pour le comportement de parole (figure 5(a)), sauf que dans ce cas, la fonction d'entrée reçoit la valeur d'activation du comportement de parole ($\mathcal{F}_{A2}(E_2(t - \Delta t))$) ainsi que les invitations aux retours ($\mathcal{I}_{BC}$). On prendra le maximum entre le produit pondéré de ces deux valeurs et une valeur d'écoute que l'on fixe à une valeur ϵ au départ.

$$\begin{aligned} E : \mathbb{R}^+ &\to [0;1] \\ t &\mapsto \max(w_{11}.(\mathcal{F}_{A2}(E_2(t - \Delta t))) \times w_{12}.\mathcal{I}_{BC}, \epsilon) \end{aligned} \tag{5}$$

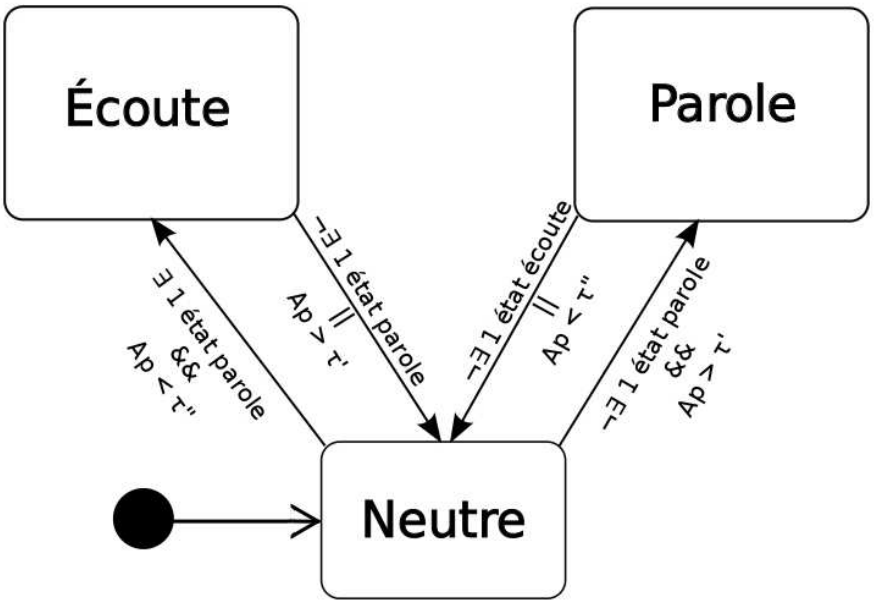


FIGURE 8 – Dynamique des états de comportement.

La sortie peut ensuite être passée à la fonction d'activation. Cette fonction est identique à celle du comportement de parole (équation 4).

La production des signaux de retour sera la valeur d'activation du comportement. En effet, comme celle-ci doit être régulière et réactive aux invitations, on aura une production égale à la valeur ϵ fixée, ou alors plus forte en cas d'invitation et de comportement de parole fort en présence.

Nous venons de décrire comment se présentait le couplage de ces deux comportements, notamment à travers l'existence d'états associés, et nous allons à présent nous arrêter sur la dynamique de la sélection d'action entre ces différents états.

3.3.3 Dynamique de la sélection d'action « parole ou écoute »

Dans le cas précédent, le problème était relativement simple : on activait ou non le comportement de parole. Ici, il faut que les deux comportements soient activés ou non de manière régulée. Il faut veiller à la bonne mise en place de leur cohabitation et nous allons détailler comment le système passe d'un état à un autre, activant tel ou tel comportement.

Pour répondre aux hypothèses de départ, il a fallu mettre en place une dynamique des changements d'états. Comme expliqué précédemment, à chaque état est associé un comportement et il faut pouvoir passer de l'un à l'autre de manière fluide et simple.

Toutes ces conditions sont résumées sur la figure 8 qui permet une visualisation plus schématique des situations décrites.

L'état central est l'état neutre. Rien ne se passe dans cet état, cependant on y passe (même pour une durée infiniment courte) entre un état de parole et un état d'écoute (et vice-versa).

Pour passer de l'état neutre à l'état de parole, on vérifie que personne n'est en état de parole (personne ne parle), et il faut que la valeur d'activation de parole de l'agent (Ap) ait dépassée un seuil τ' fixé.

Pour revenir à l'état neutre lorsque l'on est à l'état de parole, il faut que la valeur d'activation de parole soit inférieure à un seuil τ'' ou bien que personne ne soit à l'état

d'écoute. Cependant, afin d'éviter que l'état de parole soit activé, puis immédiatement désactivé, il faudra y rester un temps t.

Par conséquent, pour qu'un comportement de parole puisse exister plus d'un temps t, il faut qu'il se créée en parallèle un comportement d'écoute. On intègre donc dans le modèle le fait que le comportement de parole doit rester suffisamment longtemps activer pour qu'un comportement d'écoute ait le temps de s'activer.

Et enfin, on désactivera le comportement d'écoute lorsqu'au moins l'un des deux cas suivants sera vérifié : la valeur d'activation de parole aura atteint un seuil τ' ou lorsque plus personne ne sera à l'état de parole.

Les paramètres τ' et τ'' sont là pour illustrer la notion d'hystérésis de la transition entre comportements. En effet, dans le modèle, un comportement pourra être activé et désactivé sur des valeurs différentes représentées par ces deux paramètres.

Nous venons de présenter les principes du modèle comportemental développé, puis l'avons adapté à un comportement de parole, avant enfin de le mettre en place en cohabitation avec un comportement d'écoute bâti également sur ce modèle comportemental. Afin de pouvoir l'étudier de manière plus précise et concrète nous avons dû l'implémenter. Nous allons à présent étudier les implémentations qui ont été effectuées et présenter les résultats obtenus par simulation.

4 Implémentation du modèle et résultats

Lors du développement du modèle, afin d'observer de manière aisée les réactions de ce dernier, nous l'avons implémenté au fur et à mesure sur la plateforme Scilab. Nous allons présenter ici les comportements du modèle comportemental seul tout d'abord, puis intégré à des agents avec comportements de parole et enfin intégré à des agents avec les deux comportements détaillés auparavant. Puis nous continuerons cette partie en présentant l'implémentation faite sur la plateforme Unity3D.

4.1 Modèle formel en Scilab

Avec Scilab, ce qui était intéressant c'était d'observer les résultats en sortie du modèle grâce à différentes courbes issues des résultats. Nous allons présenter tout d'abord quelles sont les différentes entrées utilisées, et les sorties associées.

4.1.1 Scénarios

Forcer les entrées du modèle revient à forcer l'intention. Différents forçages, qu'on appellera scénarios par la suite, seront passés en entrées du système et permettront de voir le comportement du modèle. Pour ces scénarios, on souhaite que les fonctions utilisées aient certaines caractéristiques particulières. Celle qui est commune à toutes est que la fonction ι de l'intention doit respecter : $\forall t \in \mathbb{R}^+, \iota(t) \in [0; 1]$. C'est pourquoi ces scénarios sont définis de la façon suivante :

$$\iota(t) = \frac{1}{1 + \exp^{(-\alpha_1 \times (t - \eta_1))}} + \frac{1}{1 + \exp^{(-\alpha_2 \times (t - \eta_2))}} - 1 \tag{6}$$

On fixe $\eta_2 > \eta_1$ et $\alpha_1 > 0$ et $\alpha_2 < 0$. Les α_i et η_i qui vont influencer respectivement les pentes de courbes ainsi que les durées de palier.

Nous avons considéré trois types de scénarios :

1. scénario impulsion ;
2. scénario symétrique ;
3. scénario asymétrique.

Le premier correspond à une intention brève dans le temps et d'amplitude maximum. Pour ça, nous avons posé : $\alpha_1 = 21, \alpha_2 = -20$ et $\eta_1 = 12, \eta_2 = 13$. Ce qui permet une croissance et décroissance rapides, sur une durée courte (durée : $\eta_2 - \eta_1$). Ça correspondrait dans le cadre d'une conversation à une intervention brève de type réponse courte (« Oui. », « Je vois. », « Peux-tu répéter ? », etc.).

Le second correspond à un patron d'intention de montée-descente symétriques et rapides, avec un palier d'une durée significative dans le temps. Ici, $\alpha_1 = 5, \alpha_2 = -5$ et $\eta_1 = 12, \eta_2 = 30$. Ce scénario représenterait plus une intervention réfléchie et longue dans le temps de type phrase de discours (« La manipulation de cet outil nécessite des compétences particulières comme la maîtrise du langage d'implémentation lié. », etc.).

Enfin, le troisième type de scénario comprend un sous ensemble de deux scénarios puisqu'il s'agit d'une asymétrie. Il s'agit de pattern d'intention de montées et descentes asymétriques. Cette asymétrie pourrait correspondre à un volonté progressive de réalisation d'un comportement ou dans le second cas d'une volonté dégressive de la tenue d'un comportement en cours de réalisation. Nous avons posé : $\alpha_1 = 0.25, \alpha_2 = -6$ et $\eta_1 = 25, \eta_2 = 50$ pour une montée progressive et $\alpha_1 = 6, \alpha_2 = -0.25$ et $\eta_1 = 5, \eta_2 = 30$ pour une descente progressive.

Nous avons soumis ces scénarios au modèle. D'abord seul, c'est-à-dire comme si on appliquait le modèle à un comportement d'une entité seule, sans perturbation extérieure.

4.1.2 Modèle comportemental général

On a confronté ce modèle à tous les scénarios définis précédemment. Dans un premier temps, seule l'intention est forcée et par conséquent apparait seule dans le produit pondéré. Ça revient à passer directement le forçage dans la fonction d'activation 4.

Dans le premier cas on constate que le modèle suit globalement la courbe d'intention malgré le temps de cette dernière (1.5 unités de temps). figure 9(a)

Dans le cas du scénario symétrique, on constate que le modèle suit la courbe d'intention sur la totalité de son temps d'existence (environ 20 unités de temps). figure 9(d)

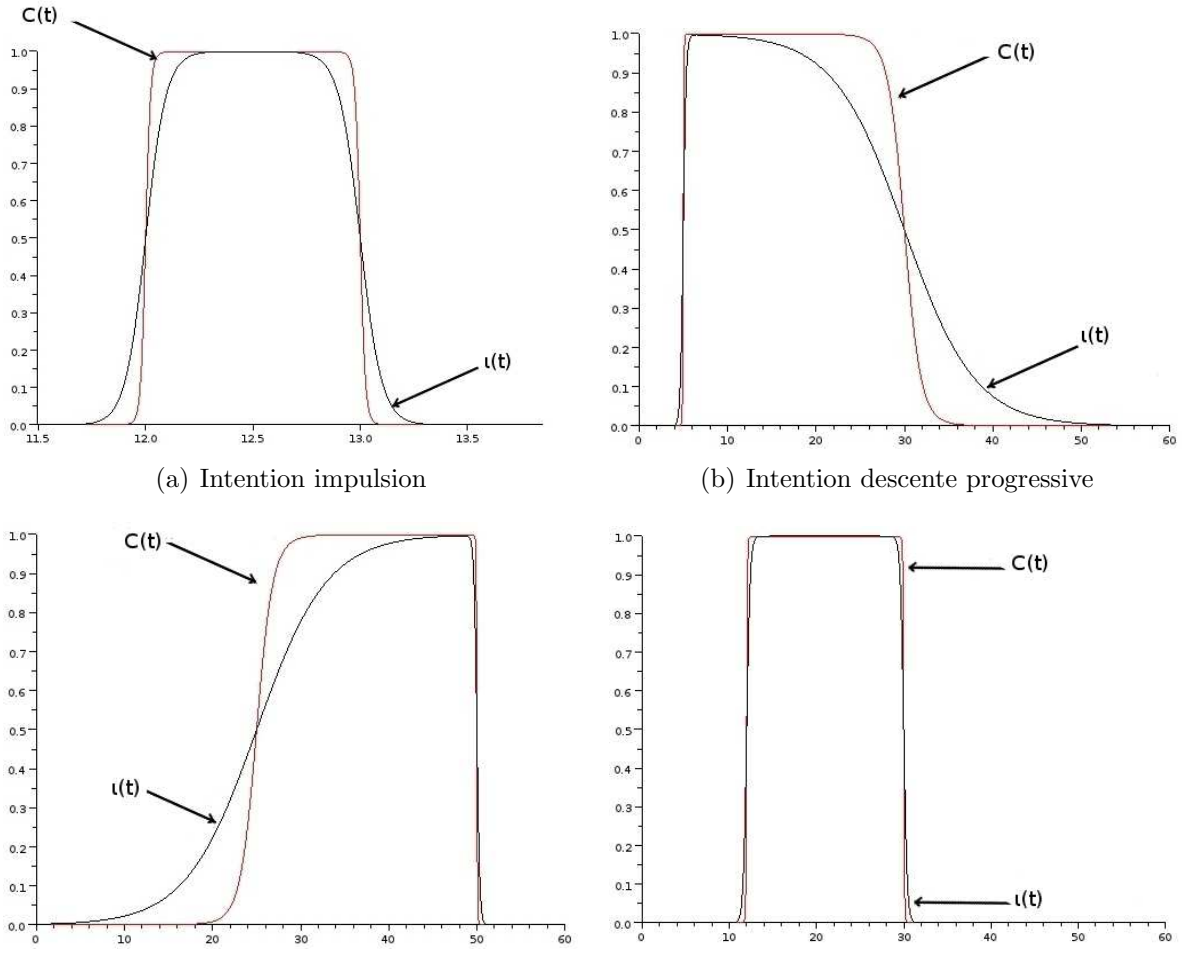
Enfin, dans les scénarios asymétriques, on constate que le modèle a une montée (respectivement descente) plus rapide que l'intention (figure 9(c) et 9(b)).

On peut noter que les courbes d'intention et de comportement se croisent à la valeur de 0.5 dans tous les scénarios et cas d'évolution de courbe (croissance de l'intention ou décroissance). Cette propriété peut être modifiée en modifiant la valeur η de la fonction d'activation suivant les besoins de modélisation.

Nous nous intéressions ici au comportement basique afin d'observer les réactions primaires du modèle. Mais l'intérêt dans notre étude est de confronter le modèle à un système impliquant deux ou plusieurs comportements. Nous allons voir à présent ce qu'il en est pour deux comportements symétriques représentant des comportements de parole.

4.1.3 Agents avec comportement de parole

On va à présent soumettre le système à la présence d'un autre agent qui va interagir, cela dans le but de simuler une conversation tenue entre deux protagonistes, mais on ne considère qu'un comportement, celui de locuteur attentif. On a vu auparavant quelles



(a) Intention impulsion

(b) Intention descente progressive

(c) Intention montée progressive

(d) Intention symétrique

FIGURE 9 – Valeur d'activation du comportement calculée en fonction du temps (C(t)). Comparaison entre intention fournie ($\iota(t)$) et valeur d'activation en sortie (C(t)).

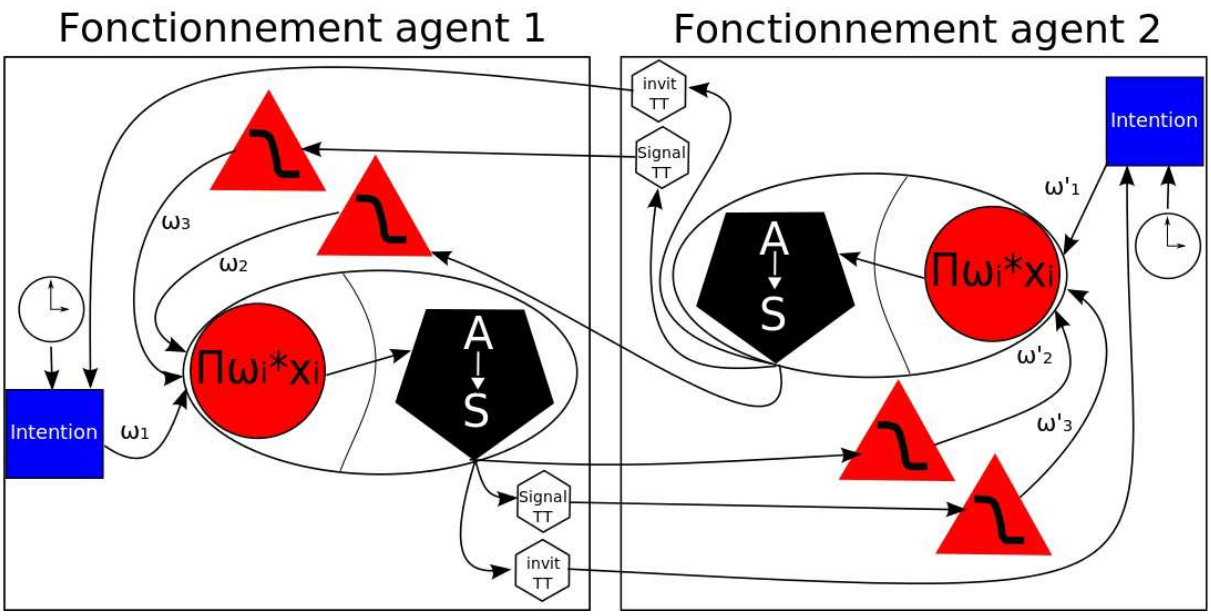


FIGURE 10 – Deux agents attentifs avec comportements de parole, et production/interprétation d'indices de tour de parole.

étaient les réactions du système en fonction de l'intention, et maintenant, nous allons ajouter des prises en compte extérieures pour simuler l'interaction avec l'autre. Ces prises en compte vont comporter l'activation précédente de l'autre pour commencer.

Pour prendre en compte l'activité de l'autre pour calculer l'activation du comportement, on va intégrer ce que fait l'autre dans la boucle de calcul.

Pour observer les réactions du modèle, nous avons simulé la présence de deux agents dits attentifs (figure 10), c'est-à-dire qui prennent en compte le comportement de l'autre comme décrit précédemment et produisant/interprétant des invitations et signaux de parole. Ces deux agents seront parfaitement identiques au niveau de l'utilisation du modèle : mêmes fonctions de calculs, mêmes poids initiaux pour le produit pondéré, seules différeront les intentions passées en entrées de chacun d'entre eux.

```
1   function [compo,invit,signal]=attentiveAgent(intention, compoAutres,
        poids,ancienCompo,invitAutres,signauxAutres)
2     indice = grand(1,1,"unf",1,5);
3     //Recuperation aleatoire du signal et de l'invitation utilises
4     valeurSignalUtilise = signauxAutres(indice);
5     valeurInvitUtilise = invitAutres(indice);
6
7     //Calcul des inhibitions (valeur activation des autres et signaux de
          prise de parole des autres)
8     compoInhib = sigexp(1,1,-17,0.5,compoAutres)
9     signauxInhib = sigexp(1,1,-17,0.5,valeurSignalUtilise)
10
11    entrees = [intention compoInhib signauxInhib];
12    //Modification du poids de prise en compte de l'intention en fonction
          de la valeur d'invitation
13    poidsIntention = poids(1);
14    poidsIntention = poidsIntention + valeurInvitUtilise;
15    poids(1) = poidsIntention;
```

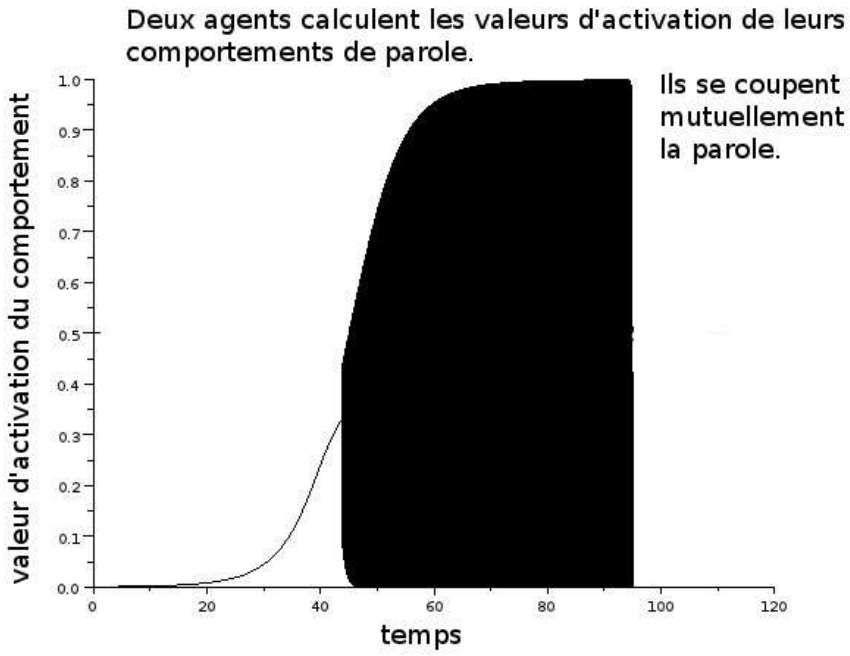


FIGURE 11 – Deux agents strictement identiques et prise en compte du dernier indice produit par l'autre uniquement. $\iota(t)$ est le scénario montée progressive pour les deux agents.

```
16
17          //Fonction combinaison
18          prod = produitPondere(entrees, poids);
19          //Fonction d'activation
20          act = sigexp(1,1,17,0.5,prod)
21
22          //Fonction de sortie calculant la valeur d'activation du comportement,
                valeur d'invitation et de signal
23          [compo,invit,signal] = compoInvitSignalTT(act,oldCompo);
24      endfunction
```

Dans notre cas, nous avons fait le choix de prendre des valeurs pour les fonctions d'activation et de désactivation (*sigexp*) telles que : $\alpha = \pm 17$ et $\eta = 0.5$.

Comme expliqué précédemment, l'interprétation des signaux et invitations se fait sur les n derniers produits. Ici, nous avons pris $n = 5$ et allons choisir aléatoirement l'un de ces 5 derniers signaux. La variable indice va servir de coefficient aléatoire pour faire ces choix.

Cette prise en compte aléatoire permet de ne pas déterminer en amont comment vont s'organiser les échanges de paroles entre les deux agents. De plus, ils permettent d'éviter pour la valeur d'activation de passer alternativement de valeur forte à valeur faible sur une durée longue. En effet, les agents produiraient exactement les mêmes comportements et seraient donc incapables de s'accorder sur un locuteur ou un auditeur, passant à l'un, puis à l'autre (figure 11).

La prise en compte des invitations de prise de parole se fait par l'augmentation de la valeur de prise en compte de l'intention de l'agent. Le poids initial étant à 1, plus l'invitation sera forte, et plus l'agent sera enclin à avoir une valeur d'activation élevée.

On peut constater que les deux agents se « disputent » la parole sur un temps très court avant que l'un d'entre eux n'arrive à prendre le pas sur l'autre (figure 12).

On se rend compte que les deux agents cohabitent et le modèle semble avoir un comportement en adéquation avec nos attentes.

Cependant, le comportement de parole seul manquerait de relief si aucun comportement d'écoute lui est associé, c'est pourquoi nous avons implémenté également ce dernier, et c'est ce que nous allons à présent détailler.

4.1.4 Agents avec comportements de parole et d'écoute

Nous avons implémenté en Scilab le comportement d'écoute, et observé son déroulement associé au comportement de parole déjà étudié.

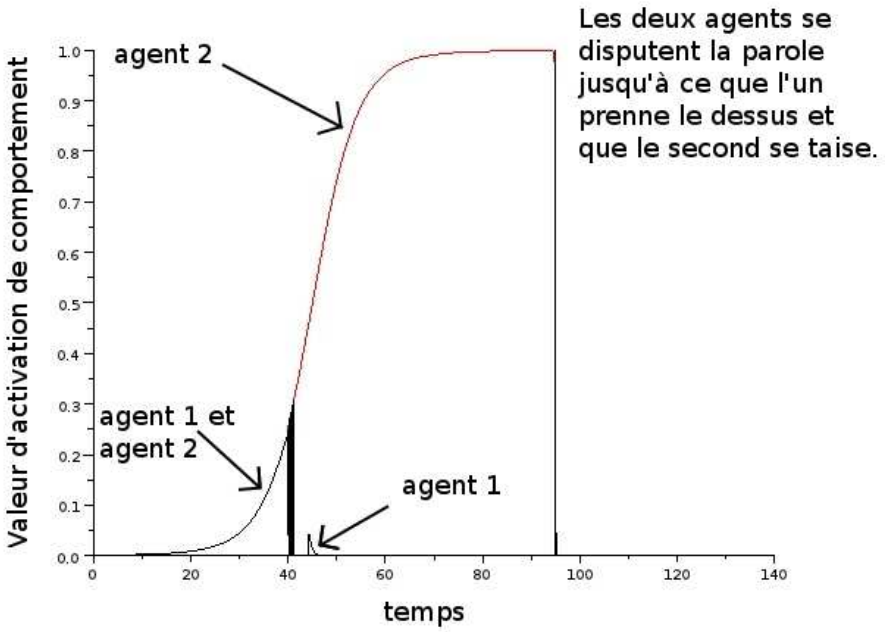
À noter que plus que le comportement d'écoute, il fallait mettre en place les 3 états discutés précédemment, à savoir : état de parole, état d'écoute, et état neutre.

```
1  function main()
2    //Declaration des variables
3    [...]
4    //Boucle d'execution du programme
5    for i=0:pas:tailleMesure
6    //Les signaux, invitations calcules au pas precedent sont ajoutes en
         fin de liste
7        [...]
8    //Stockage comportement calcule au pas precedent
9      [...]
10   //On doit connaitre le nombre d'agent actuellement en parole ou ecoute
11       nbEtatsParole = nbEtatsParole + nbAjoutP;
12       nbEtatsEcoute = nbEtatsEcoute + nbAjoutE;
13       nbEtatsParole = nbEtatsParole - nbRetireP;
14       nbEtatsEcoute = nbEtatsEcoute - nbRetireE;
15   //Remise a zero des variables de decompte de fin de boucle
16       nbAjoutP = 0; nbAjoutE = 0 ;
17       nbRetireP = 0; nbRetireE = 0;
18
19   //***Calcul de la valeur d'activation, du comportement et des signaux/
         invitations AGENT1***
20       [...]
```

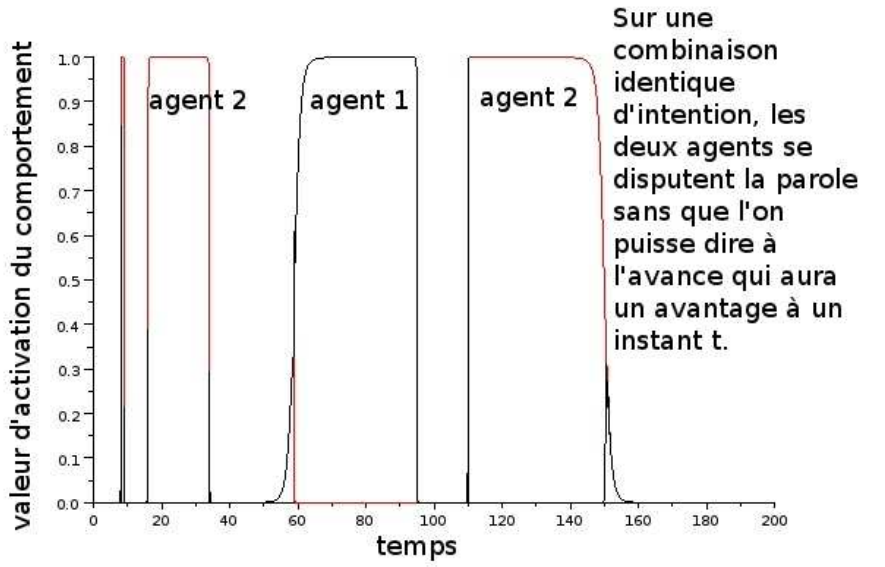
Pour éviter toute influence non désirée de changement des conditions de vie des agents, on doit mettre à jour certaines variables en début de boucle, et notamment les variables décomptant le nombre d'agent dans un état spécifique.

Ces états ont également dû être mis en place, et c'est ce qu'on a fait ici. On ne vous présente le cas que pour un agent, c'est identique pour le second à certaines variables près.

À noter le point important du temps de déclenchement de parole (*declParole1*). Comme on souhaite que l'agent reste dans un état de parole un temps minimal pour laisser aux autres agents le temps d'activer le comportement d'écoute, nous avons fixé un nombre de pas nécessaires par la variable *nbPasParole* qui permet à l'agent de rester en comportement de parole un certain temps même s'il n'y a pas d'auditoire immédiatement. La simulation durait 200 unités de temps, divisées en pas de 0.01 unité, et nous avons fait le choix de fixer ce temps à 2 unités de temps.



(a) Intention montée progressive



(b) Intentions multiples

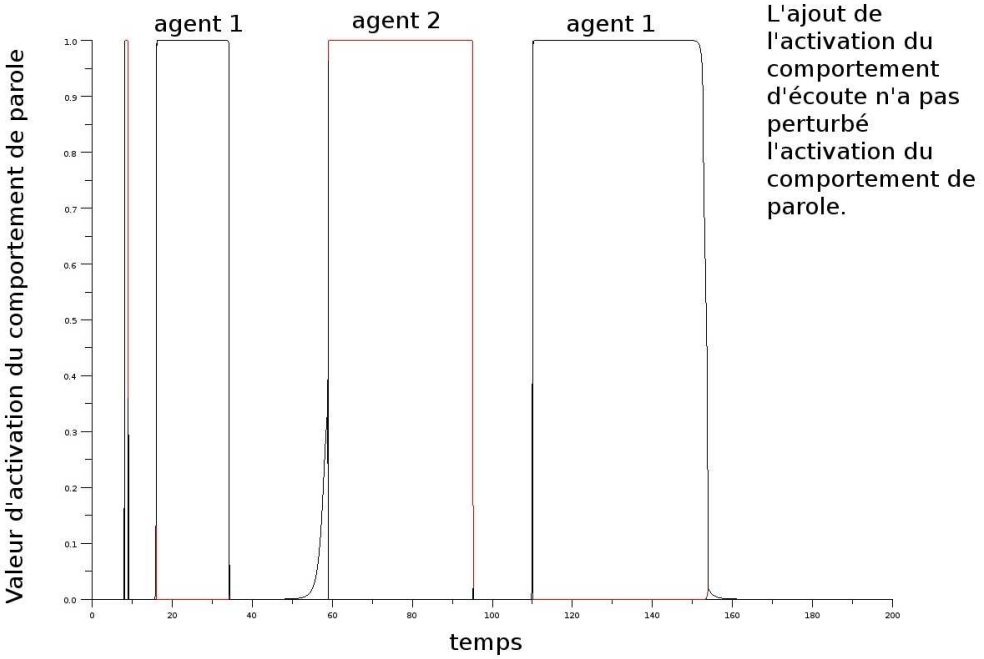
FIGURE 12 – Valeurs d'activation de deux agents avec lecture aléatoire des 5 derniers signaux/invitations sur la même intention. Un agent prend la parole, obligeant l'autre à se taire, sans que l'on puisse déterminer à l'avance qui parle et qui se tait.

```
1       //***Calcul de la valeur d'activation, du comportement et des signaux/
            invitations AGENT2***
2           [...]
3       //Changement des etats d'agent en fonction des valeurs de comportements
4          if(compoParole1 > seuilTau1 & nbEtatsParole == 0 & etatAgent1 ==
               NEUTRE)
5              etatAgent1 = PAROLE;
6              nbAjoutP = nbAjoutP +1;
7              declParole1 = i;
8          end
9          if(compoParole1 < seuilTau2 & nbEtatsParole>0 & etatAgent1 == NEUTRE)
10             etatAgent1 = ECOUTE;
11             nbAjoutE = nbAjoutE +1;
12         end
13         if((compoParole1 < seuilTau2 | nbEtatsEcoute==0) & etatAgent1 ==
               PAROLE & declParole1+nbPasParole<i)
14             etatAgent1 = NEUTRE;
15             nbRetireP = nbRetireP +1;
16         end
17         if ((compoParole1 > seuilTau1 | nbEtatsParole==0) & etatAgent1 ==
               ECOUTE)
18             etatAgent1 = NEUTRE;
19             nbRetireE = nbRetireE +1;
20         end
21      //Memes conditions pour l'agent 2, sur les variables compoParole2 et
            etatAgent2
22         [...]
23      //Stockage des points pour les resultats
24        [...]
25      //Fin de boucle
26      end
27      //Affichage resultats
28      [...]
29    endfunction
```

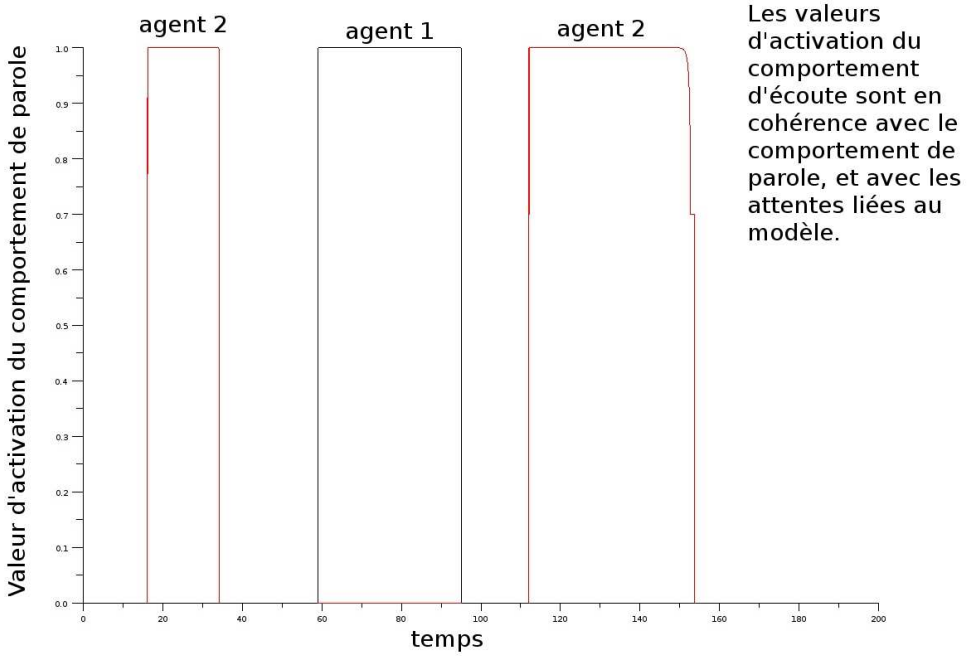
Le but de cette implémentation était premièrement de voir si les deux comportements pouvaient cohabiter. Ce que nous avons pu vérifier par cette implémentation, c'est que le modèle permet aux agents de suivre un rythme de conversation intuitif. C'est-à-dire qu'ils ne se coupaient pas sans cesse la parole, mais que pour autant ils ne se trouvaient jamais en situation de prise de parole définitive. À intentions égales, les deux agents ont un partage du temps de parole.

Le modèle reste cohérent dans cette configuration également. On a l'alternance des états suivant le comportement que nous souhaitions (figure 14) et aucun des deux agents n'a un comportement prééminent par rapport à l'autre (figures 13(a) et 13(b)). À noter que lorsque la durée de parole est trop courte (le scénario impulsion par exemple), le comportement d'écoute ne s'active pas.

Cette implémentation sur la plateforme Scilab a permis d'observer le modèle de manière théorique et calculatoire. Nous avons souhaité ensuite l'implémenter de manière simple sur des personnages animés pour observer son comportement de façon visuelle, ce qui nous permettait d'avoir un rendu plus « réel » que les observations théoriques et de calculs que



(a) Valeur d'activation du comportement de parole des agents



(b) Valeur d'activation du comportement d'écoute

FIGURE 13 – Cohabitation des comportements d'écoute et de parole.

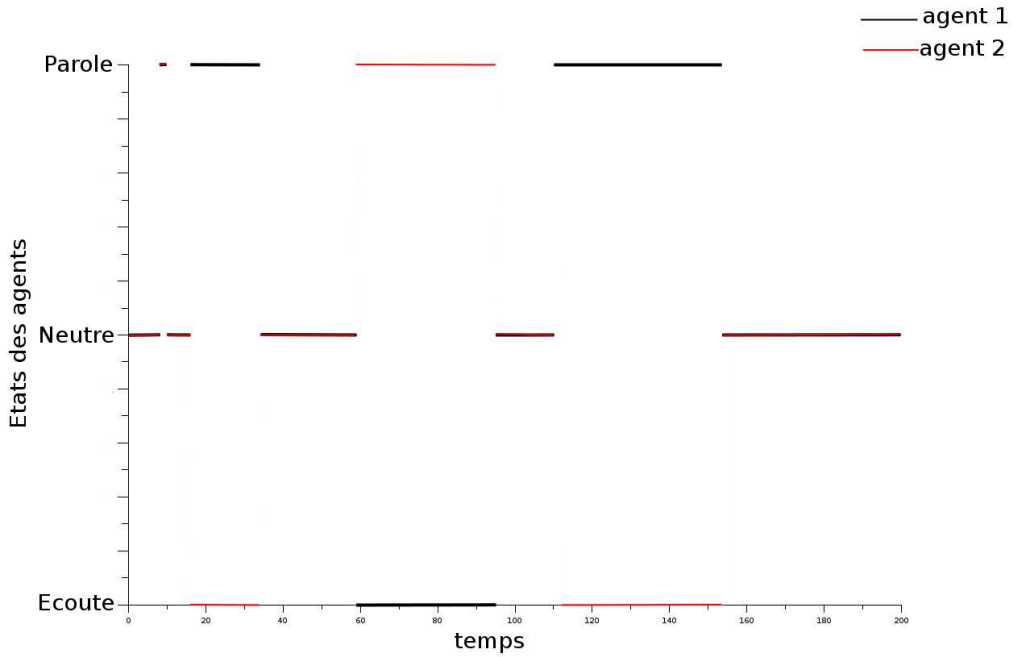


FIGURE 14 – Alternance des états de comportements des deux agents.

Scilab apportait.

4.2 Application sous Unity3D

Le moteur Unity a permis une implémentation plus visuelle par rapport à celle présentée par Scilab. Les deux approches sont différentes, et utiles à l'étude de notre modèle. Nous allons dans un premier temps présenter les principes et contraintes liés à cette plateforme, puis nous allons ensuite détailler la classe d'agent qui a été développée grâce au modèle.

4.2.1 Principes - contraintes

La plateforme Unity permet le développement d'applications de types jeux ou animation en temps réel. Elle permet l'utilisation de ressources comme l'audio ou de la vidéo ce qui nous a paru intéressant pour implémenter nos agents virtuels conversationnels. Le développement des animations se base à partir d'asset.

Cette plateforme permet l'utilisation de différents langages pour développer le code applicatif comme Javascript, Boo et C#. C'est ce dernier que nous avons choisi d'utiliser, et donc d'utiliser ses points forts tout en s'adaptant à ses contraintes.

C'est un langage objet, ce qui nous a permis le développement d'un agent en tant que tel. Cependant, le moteur Unity ne permet pas l'exécution en parallèle des comportements des agents. Ceci a pu être contraignant dans le développement puisqu'il a fallu mettre en place l'affectation de certaines variables de manière légèrement différente de celle imaginée au départ.

Ainsi, l'environnement des agents a servi de lieu d'échange de certaines informations utiles aux calculs des agents. Cela ne reste malgré tout pas illogique si on imagine que la conversation prend place dans un environnement et que cette échange a en effet lieu à cet endroit et non directement entre les protagonistes.

Pour l'implémentation sur Unity, par manque de temps, nous n'avons pas pu développer la partie écoute. C'est pourquoi par la suite, seule la partie comportement de parole sera présentée. Nous avons repris un modèle 3D existant fourni en guise d'exemple avec la plateforme Unity3D, et utilisé les objets graphiques et certaines animations qu'il contenait, notamment pour la conception des agents.

Voici donc quelles étaient les différentes contraintes auxquelles nous avons été confrontés. Cela étant, nous avons pu implémenter notre modèle sur cette plateforme, et nous allons à présent détailler les résultats qui en découlent.

4.2.2 Résultat

Les résultats de cette implémentation réside dans le code C# produit et les animations en découlant. Nous allons voir les choix qui ont été fait au niveau de ce code afin d'observer comment cette implémentation s'est mise en place.

La première chose a été de profiter de l'orientation objet du C# et de développer une classe d'agent intégrant notre modèle. Nous avons défini des attributs pour cette classe qui permettent de définir l'agent et qui seront utile au modèle.

```
1  public class Agent
2  {
3    //objet graphique de l'agent pour unity
4    private Transform graphic;
5
6    //n derniers signaux/invitations pris en compte
7    private int visionSigInv = 5;
8
9    //Variables de comportements
10   private double actualBehavior=0; private double oldBehavior = 0;
11   private List<double> invits;  private List<double> signals;
12
13   //Poids de prise en compte pour le calcul du produit pondere
14   private double intentionWeight = 0;  private double
         otherBehaviorWeight = 0;
15   private double signalsWeight = 0;
16
17   //Environnement -- autres interlocuteurs
18   private List<Agent> others;
19   [...]
20 }
```

Les valeurs qui seront utiles pour le calcul du comportement (interne à l'agent, mais également pour le calcul des autres agents) sont les attributs correspondants aux variables de comportements. Nous avons mis en place deux variables pour prendre en compte le comportement et le comportement produit au Δt précédent (*actualBehavior* et *oldBehavior*), ainsi que deux listes qui permettront de garder la production des signaux et invitations(*invits* et *signals*).

Comme le calcul de la valeur d'activation du comportement passe par un produit pondéré, nous avons intégré à la classe de l'agent les poids de ce produit. Ces poids vont représenter la « personnalité » même de l'agent, et nous avons une variable pour chaque poids (*intentionWeight* pour l'intention, *otherBehaviorWeight* pour le comportement des autres et *signalsWeight* pour les signaux).

Enfin, la classe agent intègre également le fait que les entités connaissent leurs interlocuteurs et il s'agit donc d'une liste d'autres agents (*others*).

Cet ensemble représente les attributs d'un agent, et servent à calculer un comportement en suivant notre modèle décrit précédemment. À présent, nous allons détailler la méthode associée *UpdateBehavior*.

```
1   public class Agent
2   {
3     [...]
4     public void UpdateBehavior(double intention)
5     { //Variables utiles a la methode
6       [...]
7       //Parcours des agents connus existants
8       foreach(Agent a in this.others)
9       {//Recuperation des valeurs de comportements, signaux et invitations
            des autres interlocuteurs
10        //Random pour le choix entre les n derniers signaux/invitations
11        behaviorsInhibitors += Sigexp(1,1,-17,0.5,a.OldBehavior);
12        signalsInhibitors += Sigexp(1,1,-17,0.5,a.Signals[Random.Range(0,a
              .Signals.Count)]);
13
14        invitOthers += a.Invits[Random.Range(0,a.Invits.Count)];
15      }
16
17      input.Add(intention); input.Add(behaviorsInhibitors);  input.Add(
            signalsInhibitors);
18
19      List<double> weight = new List<double>();
20      weight.Add(intentionWeight+invitOthers); weight.Add(
            otherBehaviorWeight); weight.Add(signalsWeight);
21
22      //Produit pondere
23      double prod = WeightedProduct(input, weight);
24      //Passage dans la fonction d'activation
25      actualBehavior = Sigexp(1,1,17,0.5,prod);
26    }
27    [...]
28  }
```

Cette méthode attend l'intention en paramètre. C'est cette intention qui va permettre de calculer le comportement. Tout comme l'implémentation Scilab, nous avons dû forcer cette intention et avons utilisé une combinaison des scénarios décrits auparavant.

Il s'agissait là d'implémenter la mécanique interne du modèle, avec la récupération des entrées et leur traitement. Il faut récupérer chez chaque agent de l'environnement les données qui nous intéressent, c'est-à-dire les signaux, invitations et le comportement (*a.Signals*, *a.Invits* et *a.OldBehavior*).

Une fois ces données récupérées et traitées pour certaines (passage dans la fonction de désactivation pour les signaux et le comportement), il faut ensuite que le produit pondéré (*WeightedProduct*) les prenne en charge avant de les faire passer par la fonction d'activation (*Sigexp*) qui va nous donner la valeur d'activation du comportement (*actualBehavior*).

Pour gérer la mise à jour des valeurs d'anciens comportements des agents, une méthode *update* est appelée avant de remettre à jour les deux agents. Cette méthode doit effectivement être appelée une fois que les deux agents ont été mis à jour afin de ne pas influencer les agents si l'un venait à être mis à jour avec les valeurs de l'autre incorrect.

```
1  void Update ()
2  {
3    GameObject[] othersObject = GameObject.FindGameObjectsWithTag("agent");
4    foreach(GameObject go in othersObject)
5    {
6      attentiveAgent sc = (attentiveAgent)go.GetComponent("attentiveAgent");
7      sc.GetAgent().UpdateOldBehavior();
8    }
9  }
```

La fonction *Update* permet de parcourir tous les agents existants, et de mettre à jour leur variable d'ancien comportement.

Ce modèle et ses implémentations sont obligatoirement soumis aux critiques. Jusqu'à présent, nous avons détaillé le modèle, présenté de manière concise mais aussi précise que possible les choix effectués, et nous allons maintenant soumettre le travail effectué aux discussions, mais également aux perspectives qui s'ouvrent après cela.

5 Discussions et perspectives

La conception et l'implémentation du modèle ont permis de mettre en exergue certaines de ses caractéristiques. Ici, nous allons voir les vérifications faites à propos du modèle avant de développer certains points non abordés précédemment comme le contexte multi-parties ainsi que les propriétés de notre modèle. Enfin, nous terminerons par présenter dans quelle mesure nous pouvons intégrer notre modèle au sein d'architectures d'agents conversationnels existantes.

5.1 Vérification du modèle

Actuellement, notre modèle semble tenir les objectifs qui nous étaient assignés, à savoir la robustesse et un certain réalisme dans l'exécution.

Tout d'abord il faut préciser que nous n'avions pas trouvé traces d'un tel modèle ou ressemblant dans la littérature lors de notre étude sur la problématique du tour de parole. Plusieurs réalisations ont mis en œuvre des modèles discrets états-transitions, et il nous semblait intéressant d'aborder le problème avec un modèle continu représentant le dynamisme de l'activation d'un comportement.

Dans ce contexte de stage de recherche, nous nous sommes efforcés de développer les points de base du modèle, et les simulations réalisées nous ont permis de vérifier son comportement. Le temps nous a fait défaut pour mener la moindre validation expérimentale.

Dans le futur, il faut donc prévoir une phase de validation de ce modèle.

5.2 Généralisation à des conversations multi-parties

L'étude que nous avons menée ici s'inscrit dans un projet plus large qui s'intéresse à la coordination des activités au sein d'un collectif mixte d'humains réels et virtuels. Nous

FIGURE 15 – Implémentation sous Unity3D de deux agents.

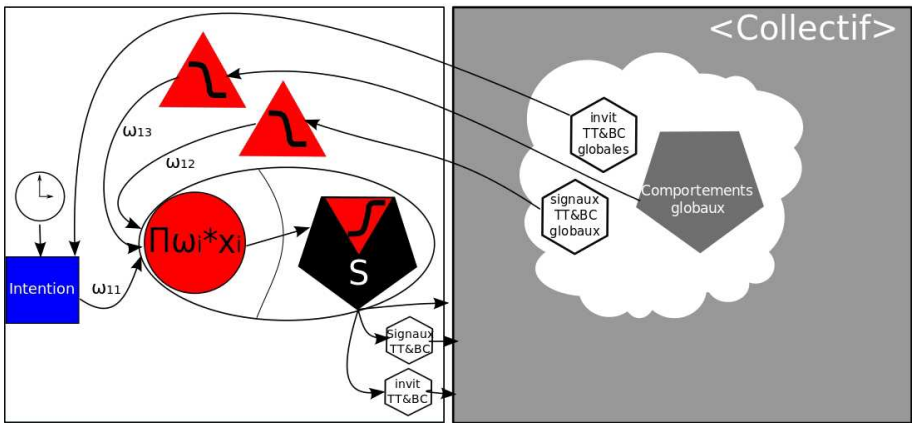


FIGURE 16 – Extension multi-parties.

avons donc conçu notre modèle afin qu'il supporte la gestion de la prise de parole pour ce type de collectif.

Cependant, la durée de travail ayant été limitée, tout n'a pas pu être étudié et seuls certains points ont été abordés.

Tout d'abord, nous avons mis en place un modèle théorique qui est généralisable au contexte des conversations multi-parties. Les implémentations ont été faites entre deux agents (néanmoins l'implémentation C# a été pensée en multi-parties). L'architecture du modèle permet d'envisager l'ajout d'autres agents. En effet, les entrées pour le calcul des comportements sont généralisables à un ensemble d'agents et pas seulement à un unique agent (figure 16). Ainsi, on peut très bien imaginer prendre le maximum des valeurs présentent dans le collectif pour calculer les valeurs d'activation des comportements.

Cependant, l'interaction avec des humains réels ou en collectif mixte n'a pas été implémentée. On suppose qu'en dehors de la transcription des comportements humains réels en données exploitables, il serait aisé de les prendre en compte pour les calculs.

5.3 Propriétés du modèle

Notre modèle a la particularité d'être continu. En effet, ses entrées sont des entrées continues comprises entre 0 et 1, et produit en sortie des valeurs également comprises entre 0 et 1.

Ce modèle s'inscrit dans un cadre de sélection de comportement, c'est-à-dire que nous avons imaginé que l'intention de l'agent était déjà créée, et que ce modèle devait apporter à ce dernier un comportement. Mais on peut également se dire de manière intuitive que les comportements (et leur sélection) ont une influence sur l'intention, et que cette intention est réciproquement très importante dans la prise de parole. Par conséquent, il serait nécessaire d'introduire une boucle de rétro-action entre la sélection du comportement et la planification de l'intention qui serait à mettre en place.

Notre modèle actuel a d'abord été pensé pour modéliser les comportements gérant les tours de parole. Cependant, il parait évident que pour s'effectuer au mieux, ces derniers dépendent des signaux et invitations. Les réalisations faites en ce sens dans notre modèle participent au bon déroulement, notamment dans le cas de conflit dans la prise de parole. Par conséquent, il pourrait être intéressant d'orienter la suite des études dans l'étude de production et interprétation d'indices afin de rendre le modèle plus réaliste.

Enfin, nous avions également en idée pour le modèle qu'il soit capable de s'auto-entretenir. C'est-à-dire que l'exécution d'un comportement contribue à son activation, ce qui rendrait la réalisation d'un comportement plus robuste à de courtes, ou faibles perturbations. Pour le cas de notre modèle, cela consisterait à ajouter par exemple en entrée, la valeur précédente d'activation du comportement calculée et de l'utiliser dans le calcul. Encore une fois, par manque de temps, tout n'a pas pu être développé en détail et il a fallu faire des choix dans la conception ainsi que dans l'implémentation.

Nous avons essayé, lors de la conception, d'imaginer l'intégration futur de ce modèle dans des architectures d'agents existantes. Nous allons à présent détailler ces dernières afin de présenter les perspectives futures.

5.4 Applications à des architectures d'agents conversationnels

Nous avons présenté le modèle, et ses propriétés et allons voir à présent comment nous imaginons les applications de notre modèle, notamment dans les architectures d'agents existantes.

Dès le départ, nous avons souhaité que le modèle soit facilement applicable à des architectures existantes et notamment l'architecture SAIBA (*Situation, Agent, Intention, Behavior, Animation*) [Kopp et al., 2006]. Il s'agit d'une structure pour les agents virtuels personnifiés (*embodied agent*) conçue dans le but d'unifier les recherches et de permettre à plusieurs entités de recherche différentes de travailler sur des modules spécifiques en étant sûr que ces derniers puissent s'intégrer dans une architecture d'agent. Dans le but de permettre la réutilisation de modules performants et d'éviter le développement d'agents à partir de zéro, cette structure se veut modulaire.

Cette architecture a été utilisée pour le développement de l'agent Greta. Cet agent conversationnel animé a été proposé par [de Rosis et al., 2003] et repris pour le projet européen SEMAINE [2]. Elle a fait l'objet de nombreux travaux de recherche et ses capacités ne cessent de s'améliorer et de s'étendre [de Sevin et al., 2010].

2. `http://www.semaine-project.eu/`

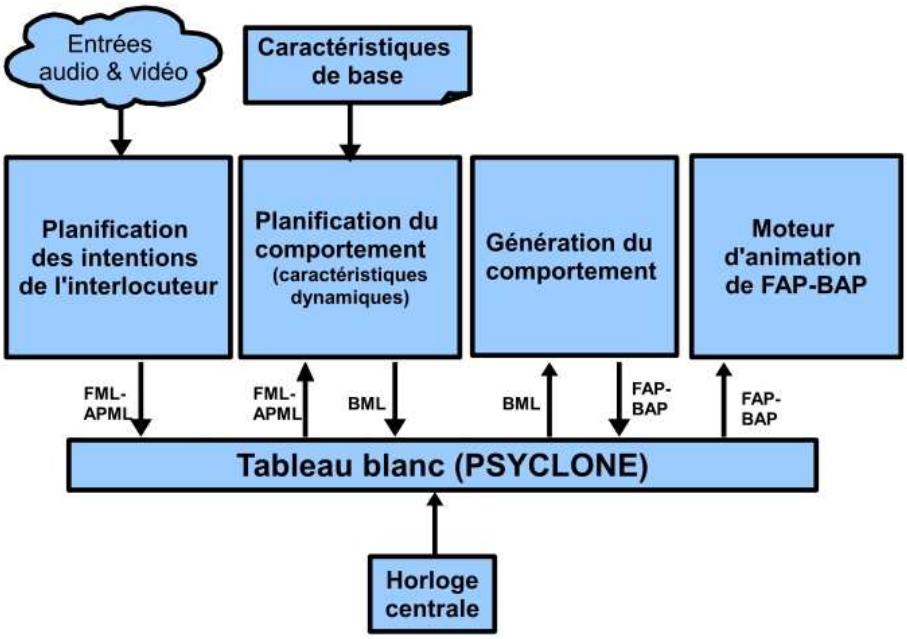


FIGURE 17 – Structure Greta [de Sevin et al., 2010].

Greta a été construite en suivant le modèle SAIBA décrit auparavant. L'architecture macroscopique (voir figure 17) est quasi identique à celle de SAIBA, à la différence près que les modules communiquent entre eux par l'ajout d'un tableau blanc que l'on retrouve également dans YTTM [Thórisson, 2002] et [Thórisson et al., 2010].

Nous avons construit notre modèle en gardant à l'esprit sa possible intégration à ce type d'architecture. Grâce aux valeurs continues qu'il utilise et retourne ainsi que son caractère dynamique, il nous semble tout à fait possible de mettre en place notre modèle dans la partie « planification du comportement » de l'architecture.

6 Conclusion

À travers ce document nous avons pu voir différentes caractéristiques du tour de parole, chez l'être humain tout d'abord, puis les modélisations qui en ont été faites. Nous avons proposé notre propre modèle comportemental, et l'avons orienté sur cette problématique de comportement conversationnel, d'abord simplement avec la parole puis en ajoutant un autre aspect qui était l'écoute.

Nous avons utilisé ce modèle dans plusieurs implémentations, mathématiques en premier lieu pour s'assurer que le développement du modèle nous intéressait et se comportait comme nous le souhaitions, puis au sein d'animations d'agent grâce à la plateforme Unity3D.

Ce modèle n'en est encore qu'à ses premiers balbutiements mais nous semble déjà intéressant et porteur de perspectives intéressantes pour la suite comme l'intégration au sein d'architectures existantes et l'étude plus approfondie des indices échangés lors de conversations.

Références

[Bevacqua, 2009] Bevacqua, E. (2009). *Computational Model of Listener Behavior for Embodied Conversational Agents*. Phd thesis, University of Paris VIII.

[de Rosis et al., 2003] de Rosis, F., Pelachaud, C., Poggi, I., Carofiglio, V., and De Carolis, B. (2003). From greta's mind to her face : modelling the dynamics of affective states in a conversational embodied agent. *International Journal of Human-Computer Studies*, 59(1–2) :81–118. Applications of Affective Computing in Human-Computer Interaction.

[de Sevin et al., 2010] de Sevin, E., Niewiadomski, R., Bevacqua, E., Pez, A.-M., Mancini, M., and Pelachaud, C. (2010). Greta, une plateforme d'agent conversationnel expressif et interactif. *Revue des sciences et technologies de l'information, RSTI–TSI*, 29 :751–776.

[Frank et al., 2009] Frank, T. D., Richardson, M. J., Lopresti-Goodman, S. M., and Turvey, M. (2009). Order parameter dynamics of body-scaled hysteresis and mode transitions in grasping behavior. *Journal of Biological Physics*, 35 :127–147.

[Gravano and Hirschberg, 2009] Gravano, A. and Hirschberg, J. (2009). Backchannel-inviting cues in task-oriented dialogue. In *Interspeech 2009*, pages 1019–1022.

[Gravano and Hirschberg, 2011] Gravano, A. and Hirschberg, J. (2011). Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3) :601–634.

[Harel, 1987] Harel, D. (1987). Statecharts : A visual formalism for complex systems. *Science of Computer Programming*, 8 :231–274.

[Jaffe and Feldstein, 1970] Jaffe, J. and Feldstein, S. (1970). Rhythms of dialogue. *Journal of Psycholinguistic Research*, 2 :369–374.

[Kopp et al., 2006] Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Thórisson, K., and Vilhjálmsson, H. (2006). Towards a common framework for multimodal generation : The behavior markup language. In Gratch, J., Young, M., Aylett, R., Ballin, D., and Olivier, P., editors, *Intelligent Virtual Agents*, volume 4133 of *Lecture Notes in Computer Science*, pages 205–217. Springer Berlin / Heidelberg.

[Kronlid, 2008] Kronlid, F. (2008). *Steps towards Multi-Party Dialogue Management*. PhD thesis, University of Gothenburg.

[Raux and Eskenazi, 2009] Raux, A. and Eskenazi, M. (2009). A finite-state turn-taking model for spoken dialog systems. In *Human Language Technologies : Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL*, pages 629–637. Association for Computational Linguistics.

[Sacks et al., 1974] Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50 :696–735.

[ter Maat et al., 2010] ter Maat, M., Truong, K., and Heylen, D. (2010). How turn-taking strategies influence users' impressions of an agent. In Allbeck, J., Badler, N., Bickmore, T., Pelachaud, C., and Safonova, A., editors, *Intelligent Virtual Agents*, volume 6356 of *Lecture Notes in Computer Science*, pages 441–453. Springer Berlin / Heidelberg.

[Thórisson, 2002] Thórisson, K. R. (2002). Natural turn-taking needs no manual : computational theory and model, from perception to action. In *Multimodality in Language and Speech Systems*, pages 173–207.

[Thórisson et al., 2010] Thórisson, K. R., Gislason, O., Jonsdottir, G. R., and Thorisson, H. T. (2010). A multiparty multimodal architecture for realtime turntaking. In *Proceedings of the 10th international conference on Intelligent virtual agents*, IVA'10, pages 350–356, Berlin, Heidelberg. Springer-Verlag.

[Yuasa and Mukawa, 2011] Yuasa, M. and Mukawa, N. (2011). Building of turn-taking avatars that express utterance attitudes. In Stephanidis, C., editor, *Universal Access in Human-Computer Interaction. Applications and Services*, volume 6768 of *Lecture Notes in Computer Science*, pages 101–107. Springer Berlin / Heidelberg.

[Yuasa et al., 2009] Yuasa, M., Tokunaga, H., and Mukawa, N. (2009). Autonomous turn-taking agent system based on behavior model. In Jacko, J., editor, *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*, volume 5612 of *Lecture Notes in Computer Science*, pages 368–373. Springer Berlin / Heidelberg.