Expressive gesture model

Quôc Anh Lê

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0036

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité “Informatique”

présentée et soutenue publiquement par

Lê Quốc Anh

le 26 juin 2013

Modèle de gestes expressifs pour un agent humanoïde

Directrice de thèse: **Catherine PELACHAUD**

Jury

M. Mohamed CHETOUANI, Maître de Conférences HDR, ISIR, Université Pierre et Marie Curie	Rapporteur
M. Rachid ALAMI, Directeur de recherches, CNRS-LAAS, Université de Toulouse	Rapporteur
M. Jean-Claude MARTIN, Professeur, CNRS-LIMSI, Université Paris-Sud	Examinateur
M. Tamy BOUBEKEUR, Maître de Conférences HDR, TSI, Télécom ParisTech	Examinateur
Mlle. Elisabetta BEVACQUA, Maître de Conférences, CERV, Ecole Nationale d’Ingénieurs de Brest	Examinateur
Mme. Catherine PELACHAUD, Directrice de recherches, CNRS-LTCI, Télécom ParisTech	Directrice de thèse

T H È S E

TELECOM ParisTech
école de l’Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

Modèle de gestes expressifs

1. Contexte et objectif de la thèse

Afin d'améliorer la communication entre une machine et l'utilisateur, on développe des agents humanoïdes qui sont les agents virtuels d'écran (ex., un agent incarné conversationnel, un agent virtuel intelligent, etc.) ou les agents physiques (ex., un robot réel, etc.). On souhaite que ces agents puissent être équipés avec pas seulement un aspect humanoïde (ex., une tête, deux mains, un corps humain, etc.) mais aussi une capacité de communication (ex., un agent communique par la parole et aussi par des comportements non-verbaux (voir la figure 1).

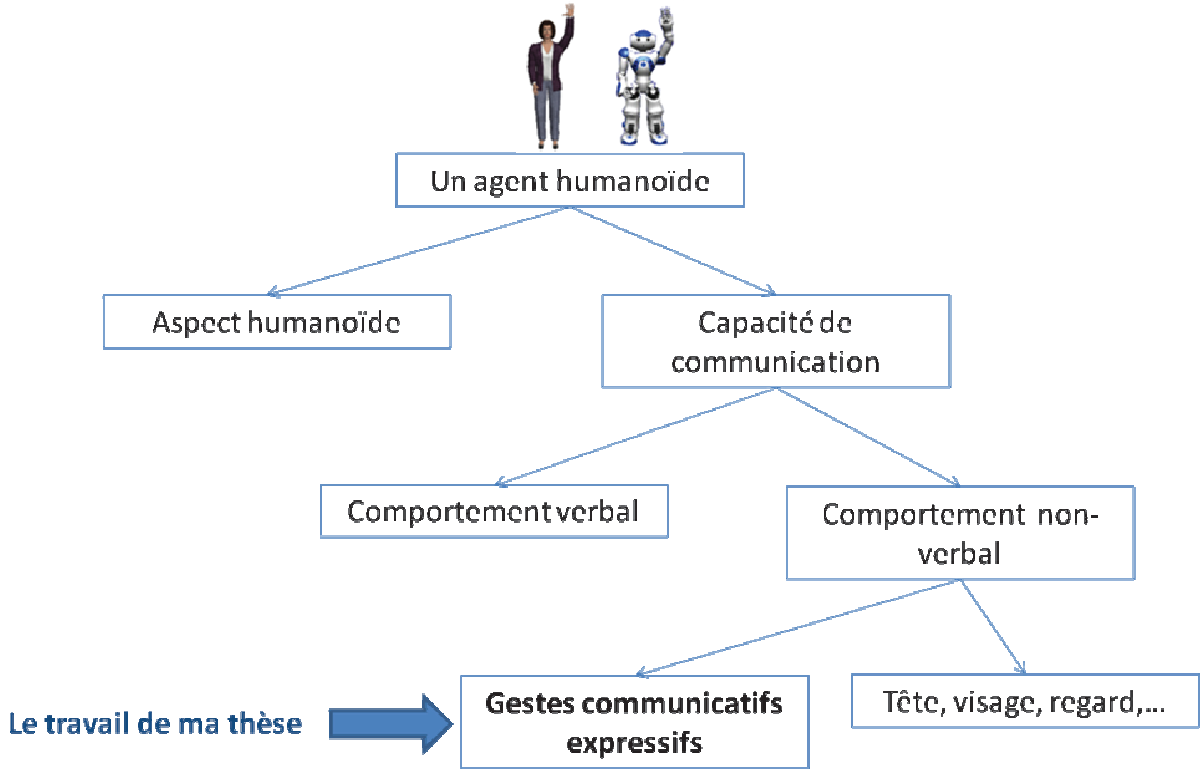


Figure 1: L'objectif de la thèse

L'objectif de la thèse est de développer des agents capables de produire des gestes expressifs communicatifs. Ce travail a été fait dans le cadre du projet ANR GVLEX (2009-2012) dont l'objectif était de doter le robot réel Nao [8] et l'agent virtuel Greta [1] de la capacité de réaliser des gestes expressifs tout en parlant. Dans ce projet, on avait quatre partenaires: 1) Aldebaran (www. aldebaran-robotics) a travaillé sur l'aspect robotique (i.e., le robot Nao); 2) LIMSI (www.limsi.fr) a travaillé sur l'aspect linguistique et sur des annotations des gestes des conteurs humains; 3) Acapela a travaillé sur une synthèse vocale pour le langage français; 4) et Télécom-

ParisTech, nous avons travaillé sur des comportements non-verbaux en général et des gestes expressifs communicatifs en particulier.

Plutôt que de développer un nouveau modèle de comportent nous nous appuyons sur un modèle existant. Depuis plusieurs années des travaux ont été menés pour doter les agents virtuels de capacité expressive. Notre approche utilise la plateforme d'agent conversationnel animé GRETA [Pelachaud, 2008].

Ainsi nous nous proposons de contrôler le comportement nonverbal du robot par un langage symbolique. L'idée est d'utiliser le même langage de représentation pour l'agent virtuel et l'agent physique, ici le robot NAO. Cela nous permet d'une part de contrôler le comportement du robot par le système GRETA et de moduler son exécution en vue de le synchroniser avec la parole et de le rendre plus expressif.

Le système GRETA calcule le comportement non-verbal que l'agent doit montrer pour communiquer un texte d'une certaine manière. Les gestes de l'agent sont stockés dans une librairie de comportements, appelée Lexicon. Ils sont décrits par une représentation symbolique. La sélection et la planification des gestes sont basées sur les informations qui enrichissent le texte d'entrée. Une fois sélectionnés, les gestes sont synchronisés avec la parole, puis ils sont réalisés. Pour calculer leur animation les gestes sélectionnés sont transformés en keyframes où chaque keyframe contient les valeurs des articulations de l'agent, et la vitesse du mouvement. L'animation de l'agent est spécifiée par des scripts décrits avec le langage de représentation Behavior Markup Language BML [Kopp et al, 2005]. Comme le robot et l'agent virtuel n'ont pas les mêmes capacités, les scripts doivent être fournis de façons utilisables par les deux agents. Les travaux de la thèse se concentrent principalement sur l'animation du robot Nao et de l'agent virtuel Greta. En détails, ils sont :

i) Développer un réalisateur de comportements dont les informations d'entrée sont les descriptions des comportements encodées avec langage de représentation BML.

ii) Augmenter l'expressivité des gestes en ajoutant les paramètres de dimensions gestuelles tels que l'extension spatiale, l'extension temporelle, la fluidité, la force et la répétition [26].

iii) Elaborer les répertoires des gestes expressifs qui sont utilisables par le robot et par l'agent virtuel. Le langage BML devra être étendu pour encoder les descriptions de ces gestes.

iv) Evaluer le système implémenté pour 1) s'assurer que les deux agents (physiques et virtuels) transmettent des informations similaires pour un ensemble d'intentions, 2) vérifier et comparer la capacité des deux agents à lire une histoire expressivement (i.e. Nao et Greta).

Cependant plusieurs issues doivent être abordées :

1. Les deux systèmes d'agents, virtuels et physiques, n'ont pas les mêmes degrés de liberté. Etant donné des intentions et états émotionnels à transmettre, le système Greta calcule un ensemble de comportements. Il faut donc le robot et l'agent virtuel communiquent des informations similaires mais pas forcement en utilisant des comportements identiques.

2. La synchronisation du comportement verbal et nonverbal est une propriété essentielle. De même que pour le point précédent, le robot et l'agent virtuel ont des propriétés physiques très différentes. Le robot est une entité physique avec une masse corporelle, des articulations physiques avec une limite de vitesse et de déplacement. Ce n'est pas le cas de l'agent virtuel. Un mécanisme de synchronisation pour le comportement du robot sera développé en tenant compte de ses caractéristiques physiques.

3. L'expressivité gestuelle a plusieurs fonctions telles que transmettre des émotions [17, 18], attirer l'attention du locuteur or contraster des éléments [19]. Elle se traduit par un ensemble de dimensions, l'ampleur des comportements, leur vitesse et puissance d'exécution, leur fluidité et leur rapidité. Nous implémenterons un tel modèle pour le robot NAO.

2. Définitions et rôles des gestes communicatifs expressifs

Les gestes communicatifs sont des mouvements des mains et des bras synchronisés avec la paroles; Ils sont porteurs d'informations nécessaires à la communication (McNeill, 1992; Gullberg, 1998, Butcher et Goldin-Meadow, 2000).

D'après Poggie (2008), un geste communicatif peut être représenté par la paire (signal, sens):

- Le signal est décrit par la forme et le mouvement de la main et du bras

- Le sens représente une image mentale ou une proposition qui est véhiculée par le signal.

Les gestes communicatifs ont des rôles importants pour tous les deux, le locuteur et l'interlocuteur. Pour le locuteur, il l'aide à formuler sa pensée (Krass, 1998). Pour l'interlocuteur, il fournit des informations qui peuvent être aussi bien complémentaires que redondantes, voire même contradictoires (Iverson, 1998).

Les gestes expressifs reflètent des états affectifs (Gallagher et Frith, 2004). L'expressivité correspond à la manière dont le geste est exécuté (Wallbott et al., 1986). Les rôles des gestes

expressifs sont d'attirer l'attention, de persuader les auditeurs (Chafai et Pelachaud, 2008; Kipp et Martin, 2009), ainsi que d'indiquer les états émotionnels (Wallbott et al., 1986, 1998).

3. Revue de la littérature sur le sujet de gestes expressifs

Selon le dictionnaire Le Robert, le geste est défini par un « Mouvement du corps (principalement des bras, des mains, de la tête), révélant un état d'esprit ou visant à exprimer, à exécuter quelque chose ». Cette définition ne fait aucun lien entre la parole et les gestes. Elle n'est pas spécifique aux gestes communicatifs. Dans cette thèse, on se concentre sur les gestes co-verbaux des bras et mains ainsi que de la tête. Les gestes du visage (i.e. expressions faciales) et de la posture ne sont pas étudiés ici. Notre modèle s'appuie principalement sur les travaux théoriques de la communication gestuelle d'Adam Kendon (2004), de David McNeill (1992) et de Geneviève Calbris (1983).

Les sections suivantes abordent la hiérarchie gestuelle développée par Kendon, la classification des gestes et le codage descriptible de la forme d'un geste proposé par McNeill, ainsi que la taxonomie des gestes de Calbris. La relation de synchronisation entre la parole et les gestes est une partie importante ; elle sera présentée par les observations des chercheurs listés au-dessus et des autres chercheurs.

3.1. La hiérarchie gestuelle

Selon la hiérarchie de Kendon (2004, p.108-126), une action gestuelle peut être divisée en plusieurs phases de mouvement, dans laquelle la phase obligatoire est appelé *stroke (apogée)* - elle transmet la signification d'un geste. Le *stroke* peut être précédé par une phase préparatoire qui met les articulations corporelles (i.e. la main et le poignet) à la position où aura lieu le *stroke*. Il peut être suivi par une phase de rétraction qui apporte les articulations au point de départ ou à la position initiale du geste suivant. La combinaison de la phase préparatoire et de la phase *stroke* est appelé *phrase gestuelle*. Une phrase gestuelle peut éventuellement avoir des moments dans lesquels les articulations sont tenus (i.e. holds) avant et après le *stroke ;* Ces phases permettent d'attirer l'attention des interlocuteurs et de synchroniser la parole et les gestes. Une *unité gestuelle* est définie comme une série de phrases gestuelles qui se suivent l'une après l'autre et sont terminées par une phase de rétraction. En fait, la phase de rétraction n'est pas considérée comme une partie de la phrase gestuelle, bien qu'elle appartienne à l'unité gestuelle qui contient la phrase gestuelle.

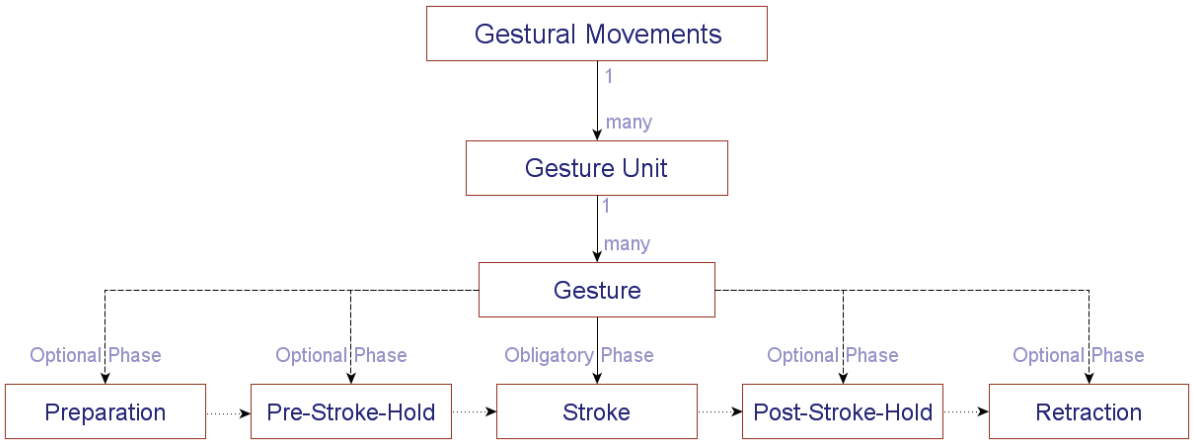


Figure 2. La hiérarchie gestuelle

3.2. Classification des gestes

Les gestes communicatifs sont classifiés suivant la taxonomie de McNeill (1992, p.12-18). Il y a quatre types de gestes : 1) les gestes *iconiques* représentent une idée ou un objet concret. Par exemple les doigts des deux mains forment un cercle lorsqu'on parle de la pleine lune ; 2) Les gestes *métaphoriques* représentent plutôt une idée abstraite qu'un objet ou événement concret. Un exemple est lorsque le locuteur utilise ses mains ouvertes et enlevées lorsqu'il dit « c'est mon idée. » ; 3) Les gestes *déictiques* sont les mouvements corporels de pointage identifiant un objet concret ou abstrait dont on est en train de parler. 4) Les gestes *bâtons* sont des mouvements de bras ou des mains synchronisés avec la parole. Souvent ils coïncident avec les syllabes accentuées de la parole. Un propriété distinct des *bâtons* est qu'ils n'ont que deux phases de mouvements (typiquement haut/bas) par rapport aux types iconiques et métaphoriques qui ont normalement trois phases (préparation-stroke-rétraction). Une autre caractéristique différente de ces gestes est que tandis que les gestes déictiques, iconiques et métaphoriques accompagnent une parole qui parle souvent d'une proposition (e.g. un objet, une idée, une position, etc.); la parole avec qui les bâtons accompagne n'en donne aucun.

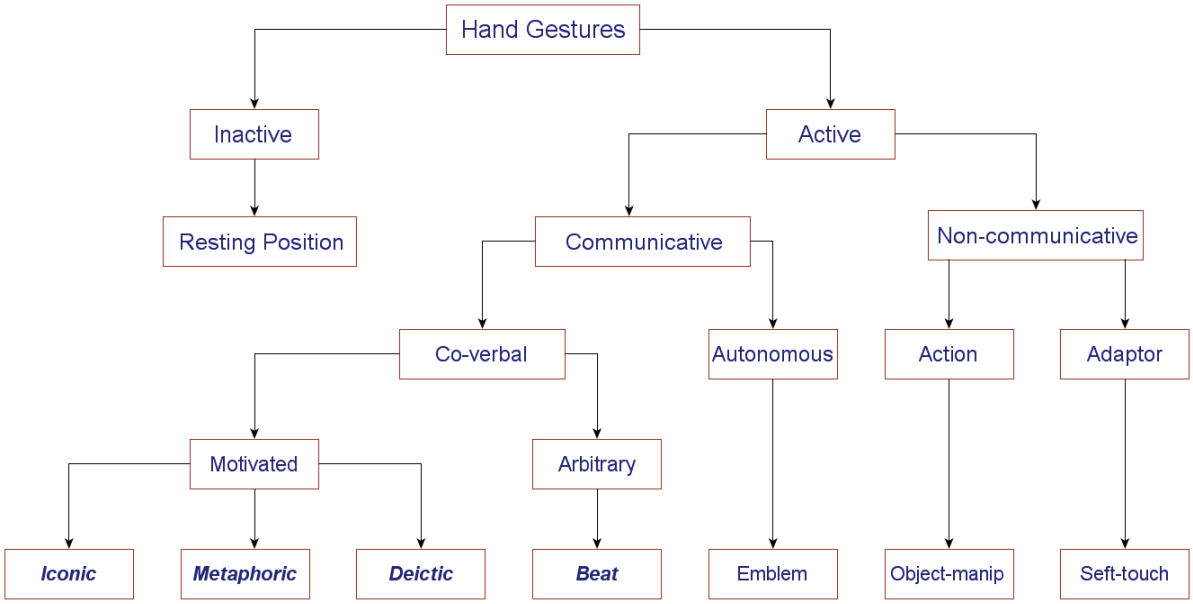


Figure 3. La classification gestuelle

3.3. La relation entre les gestes et la parole

A partir des observations que la phase *stroke* du geste coïncide avec ou juste avant les syllabes accentuées de la parole, Kendon (2004, p.127-157) conclut que les gestes ne sont pas inspirés de la parole, mais plutôt que les gestes et la parole viennent d'une même origine commune. C'est à dire que les gestes et la parole sont les deux aspects de même processus dans la quel les gestes se produisent légèrement avant la parole. Il y a une adaptation mutuelle de leur production : i) La performance des gestes est adaptée à la structure du discours. Par exemple les mouvements sont arrêtés, i.e. les mains et les bras maintiennent leurs positions; pendant que la phrase (du discours) entre parenthèse est parlée; et puis les mouvements sont repris. Un autre exemple est qu'un maintien des articulations (post-stroke-hold) peut être ajouté pour le geste couvre toute la phrase soulignée; ii) La performance de la parole est adaptée à l'exigence de l'expression gestuelle dans temps réel. Par exemple, la parole attend un instant pour que la phase préparatoire du geste rattrape la parole afin que le phase *stroke* coïncide la phrase accentuée de la parole.

McNeill (1992, p.24-25) introduit le phénomène d'anticipation gestuelle. Ce phénomène a été confirmé par une recherche récente de Ferré, 2010 [23]. L'anticipation a lieu dans la phase préparatoire. La durée de réaliser la phase préparatoire doit être prévu avant d'exécuter pour que le stroke puisse se produire au même temps avec la phrase accentuée de la parole. McNeill (1992, p.26-35) propose trois règles de synchronisation. Ces règles montrent comment les gestes et la parole sont synchronisés. Les deux premières, *règle synchrone sémantique* et *règle*

synchrone pragmatique, spécifient que si les gestes et la parole se coproduisent, ils doivent présenter les mêmes informations sémantiques, ou effectuer la même fonction pragmatique. La troisième règle dit que le phase *stroke* du geste précède, ou coïncide avec, mais ne suit pas, la syllabe accentuée de la parole.

En conclusion, les gestes et la parole sont deux aspects de l'énonciation, l'aspect imaginé et l'aspect linguistique. Ils ont une relation constante dans le temps pour transmettre en synchronisant le même contenu. Le locuteur adapte le temps de ses gestes en ajustant une tenue (i.e. hold) avant ou après le stroke pour assurer la synchronisation avec la parole.

3.4. Spécification et codage des gestes

Les gestes sont spécifiés par plusieurs paramètres : la forme de la main, l'orientation de la paume et du poignet, la forme de la trajectoire, la direction du mouvement, et la position des mains dans l'espace gestuel (McNeill 1992, p78-89). Cet espace gestuel est défini comme un système de carrés concentriques centrés sur l'acteur. Dans ce schéma (voir Figure 1), il est divisé en petits secteurs dont chacun peut être la cible pour la position du bras. McNeill a trouvé empiriquement que les gestes sont exécutés principalement dans ces secteurs.

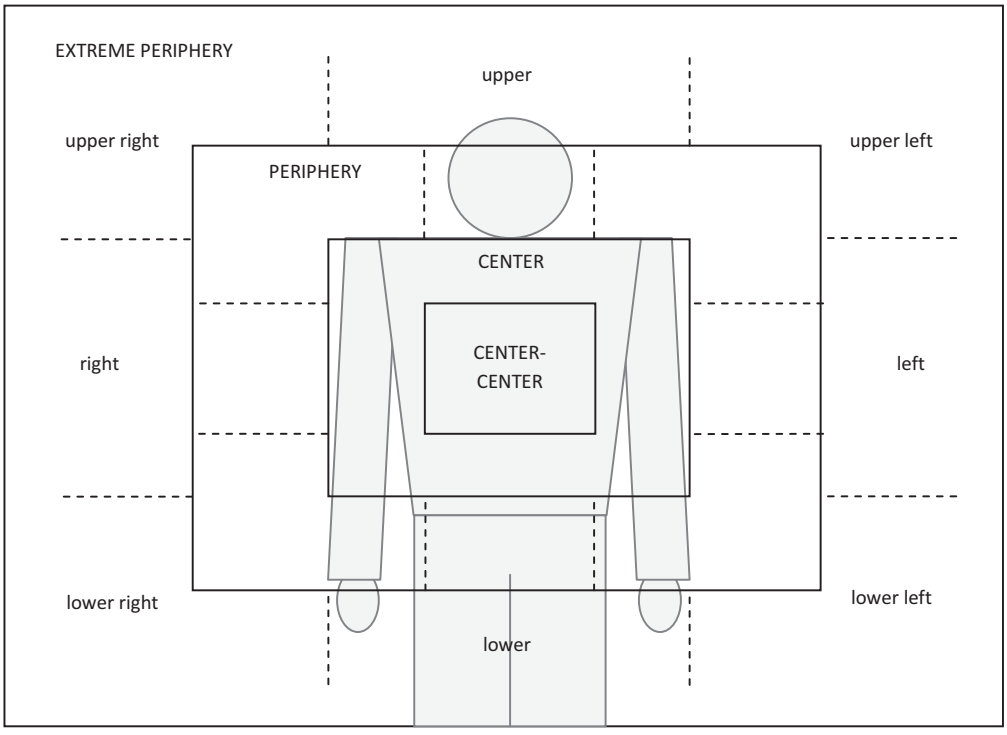


Figure 4. Carrés concentriques de Mc Neill (1992).

3.5. Variantes des gestes

Calbris (1983) a étudié dans sa thèse un répertoire de familles des gestes avec variantes. Chaque famille gestuelle englobe plusieurs cas de comportements, qui peuvent se différencier en forme, mais véhiculer un message similaire. Par exemple on a huit variantes gestuelles pour la famille de négation : « mouvement transversal répété, autrement dit secouement latéral de la tête (…), de la main (…), de l'index (…). Mouvement transversal simple de la main en plan horizontal (…) ou en plan frontal (…). Le mouvement transversal n'est pas nécessaire : pour arrêter, la main est brusquement avancée à l'horizontale (…) ou bien levée, paume contre l'extérieur (…) . Enfin, substitut de la main, l'index est levé contre l'extérieur en signe d'opposition (…) » (Calbris 1983, p.398) .

Le travail de Calbris est utile pour élaborer les librairies des gestes pour les agents qui ont capacités différentes tels que le robot Nao et l'agent virtuel Greta. Ils peuvent utiliser un élément d'une famille de geste pour transmettre un même message, même si les gestes ils sont différents dans leur forme.

4. Les travaux récents sur le contrôle des gestes expressifs de robot humanoïde

Plusieurs initiatives ont été proposées récemment pour contrôler les comportements d'un robot physique humanoïde. Salem et al. (2013) utilisent le moteur de gestes de l'agent virtuel Max pour contrôler le robot humanoïde ASIMO. Holroyd et al (2011) ont mis en place un système suivant une architecture évènementielle pour résoudre le problème d'imprévisibilité de la performance de leur robot humanoïde Melvin. Ng-Thow-Hing et al (2010) développent un système qui prend un texte quelconque et puis sélectionne et produit les gestes correspondants à réaliser par le robot ASIMO. Shi et al (2010) proposent un système qui produit les comportements pour un robot correspondant des informations reçues de l'environnement. Nozawa et al. (2005) dotent leur robot de capacités de production des gestes déictiques lorsque le robot donne une présentation sur l'écran.

(a) **BANDI-II** Mead et al. (2010)

(b) **ASIMO** Salem et al. (2012)


(c) **MELVIN** Holroyd et al. (2012)


(d) **BERTI** Bremmer et al. (2009)

Ces systèmes ont plusieurs caractéristiques communes. Par exemple, ils calculent les paramètres d'animation du robot à partir d'une description symbolique des comportements encodée avec un langage de représentation tels que BML (Holroyd et al., 2011), MURML (Salem et al., 2012), MPML-HR (Nozawa et al, 2005), « Simple Communicative-behavior Markup

Language » (Shi et al., 2010). La synchronisation des gestes avec la parole est assurée en adaptant les mouvements des gestes à la structure de la parole. C'est aussi la méthode utilisée dans notre système. Certains systèmes sont dotés avec un mécanisme de rétroaction pour recevoir et traiter les informations en retour (i.e. feedback) du robot en temps réel. Les informations en retour sont utilisées pour améliorer les mouvements gestuels ou pour sélectionner un action suivante , ou pour synchroniser les gestes avec la parole.

Notre système se différencie par rapport aux travaux proposés ci-dessus. Il suit une architecture standard de génération des comportements pour un agent conversationnel animé (i.e. SAIBA [Kopp et al., 2005]). Les répertoires des gestes du système sont considérés comme un paramètre de personnalisation pour l'usage externe. En modifiant ce paramètre, nous pouvons changer les comportements ainsi qu'adapter les prototypes gestuels aux contraintes spécifiques de l'agent sans intervenir dans les codes sources du programme. De plus, dans notre système, l'expressivité des gestes est augmentée en ajoutant les paramètres de dimensions gestuelles tels que l'extension spatiale, l'extension temporelle, la fluidité, la force et la répétions du mouvement .

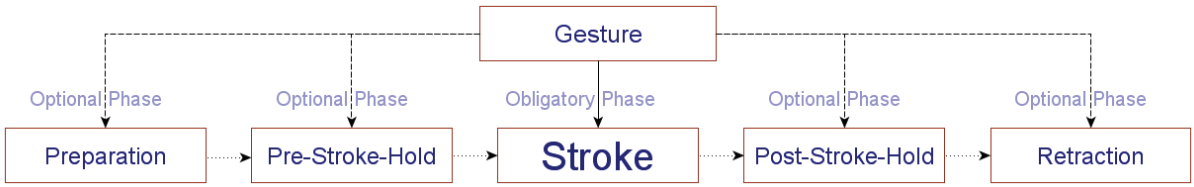
Comme le système MAX qui est utilisé pour deux agents qui ayant des capacités gestuelles différentes (Max vs. ASIMO), notre système est utilisé pour contrôler l'agent virtuel Greta et le robot physique Nao. Cependant, en divisant le système en deux parties séparées lors du calcul des paramètres d'animation (une partie commune et l'autre spécifique), nous pouvons appliquer notre système à un nouvel agent en réutilisant la plupart des modules de base.

5. Questions de recherches

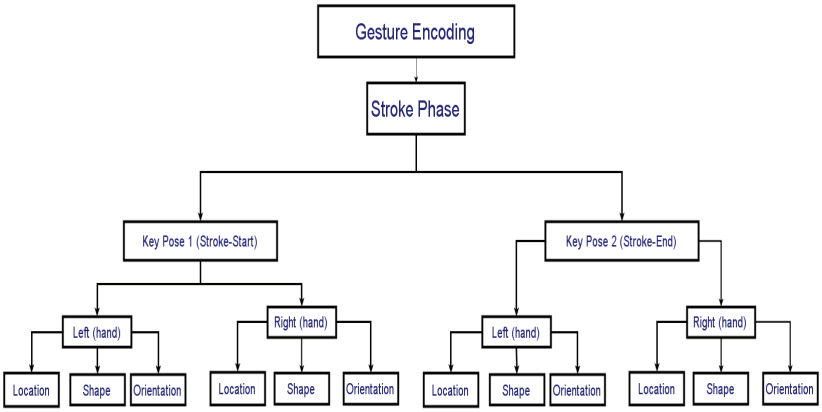
L'objectif de ma thèse est de développer un modèle computationnel des gestes expressifs. Pour la faire, on s'est basé sur des études théoriques des gestes humains est les applique à un agent humanoïde. Il y avait six questions de recherche à poser dans cette thèse. Elles sont:

1. Comment représenter les gestes humains?
2. Comment déterminer la vitesse des gestes?
3. Comment synchroniser les gestes avec la parole?
4. Comment déterminer l'enchainement des gestes dans un discours (i.e., la coarticulation des gestes)?
5. Comment les rendre expressifs?
6. Comment conduire des testes expressifs pour valider notre approche?

5.1. Concernant la première question, il y a deux exigences afin de représenter les gestes humains: i) Un geste humain doit être encodé en utilisant suffisamment d'informations pour reconstruire ce geste et le réaliser par un agent sans en perdre la signification; ii) La description du geste devrait rester à un niveau d'abstraction pour que la même syntaxe gestuelle puisse être utilisée pour différents types d'agents (i.e., virtuel ou physique).

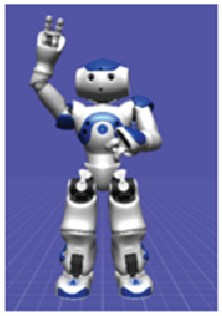


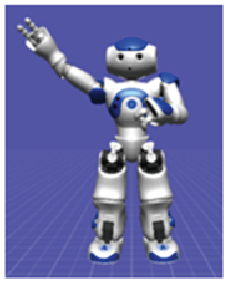
Selon des études de McNeill (1992) et Kendon (1980), un geste peut être divisé en plusieurs phases différentes (i.e., préparation, stroke, hold, rétraction) dans lesquelles la phase de stroke est la phase plus importante parce qu'elle porte la signification du geste. C'est pourquoi seul, la phase de stroke est encodée et les autres phases seront calculées automatiquement en temps réel par notre modèle.



Les gestes sont décrits symboliquement afin d'être utilisé pour différents types d'agents. Dans notre modèle on a défini un ensemble des propriétés gestuelles tels que la forme de la main, l'orientation de la paume, l'orientation des doigts, la position du poignet, la trajectoire des mouvements et le symétrie des mains. Par example:

```
1. <gesture lexeme="hello-waving" mode="RIGHT_HAND">
2. <phase type="STROKE-START" TrajectoryShape="LINEAR" >
3.         <hand side="RIGHT">
4.             <VerticalLocation>YUpperPeriphery</VerticalLocation>
5.             <HorizontalLocation>XPeriphery</HorizontalLocation>
6.             <FrontalLocation>ZNear</FrontalLocation>
7.             <HandShape>OPEN</HandShape>
8.             <PalmDirectation>AWAY</PalmDirectation>
9.         </hand>
10. </phase>
11. <phase type="STROKE-END"TrajectoryShape="LINEAR" >
12.         <hand side="RIGHT">
13.             <VerticalLocation>YUpperPeriphery</VerticalLocation>
14.             < HorizontalLocation>XExtremePeriphery</HorizontalLocation >
15.             <FrontalLocation>ZNear</FrontalLocation>
16.             <HandShape>OPEN</HandShape>
17.             <PalmDirectation>AWAY</PalmDirectation>
18.         </hand>
19.</phase>
20.</gesture>
```





5.2. Pour la deuxième question de recherche "comment déterminer la vitesse des gestes?", il faudrait définir la vitesse des gestes dans deux conditions au moins: i) la vitesse des gestes dans une condition "neutre"; et ii) la vitesse des gestes suite à un état émotionnel (colère, joie, tristesse, etc.).

On a utilisé la loi de Fitts (Fitts, 1992) afin de simuler la vitesse dans une condition "neutre" et utilsé un agent-dépendent méthode pour déterminer les vitesse "minimale" et "maximale" des gestes.

La loi de Fitts est représenté par la formule: MT = a + b*log(D/W +1)

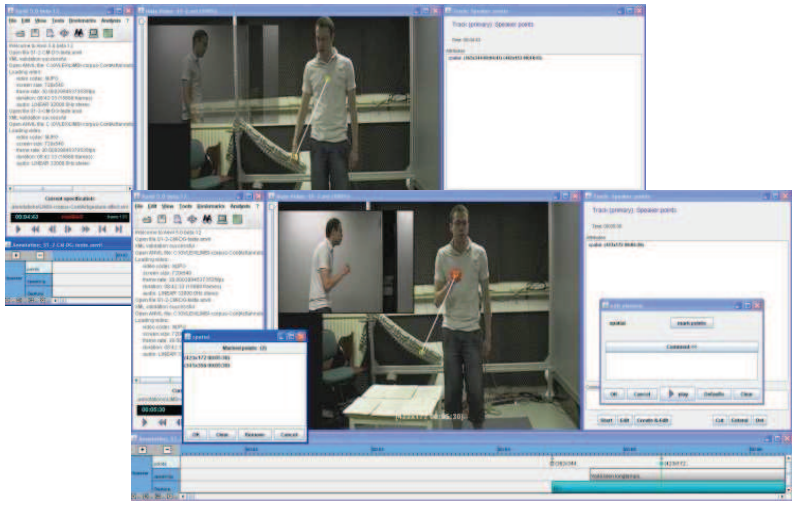
Dans laquelle, MT est le temps moyen pris pour effectuer le mouvement de la main; a et b sont déterminés empiriquement; D est la distance séparant le point de départ de la cible; W est la tolérance de la position finale.

L'indice de difficulté est calculé par la formule: ID = log(D/W+1)

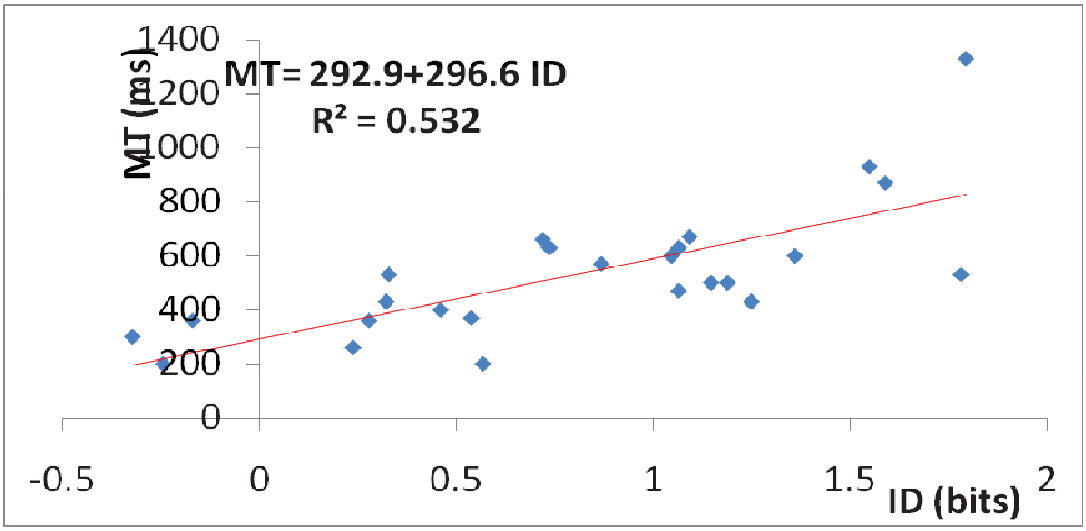
Et l'indice de performance est calculé par la formule: IP = ID/MT.

On a déterminé les paramètres a et b de la loi de Fitts par régression linéaire en utilisant les gestes des sujets humains.

ID	D	MT (ms)	IP
1.36	51.36	600	2.27
-0.25	16.88	200	-1.22
1.06	41.84	470	2.27
0.23	23.58	260	0.91
0.53	29.03	370	1.45
1.79	69.37	1330	1.35
1.18	45.59	500	2.38
1.14	44.30	500	2.29
1.78	68.77	530	3.36
0.27	24.25	360	0.75
0.32	68.48	430	0.74
0.73	9.24	630	1.15
1.06	25.69	630	1.68
0.86	30.98	570	1.50
0.46	15.80	400	1.15
0.71	42.94	660	1.07
1.04	35.62	600	1.73
0.32	15.76	530	0.60
-0.32	37.32	300	-1.06
1.58	81.16	870	1.81
1.54	49.05	930	1.65
0.56	23.26	200	2.80
-0.17	17.63	360	-0.47
1.09	8.83	670	1.62
1.24	40.53	430	2.88



Et on a construit la régression linéaire pour la loi de Fitts comme suivante:

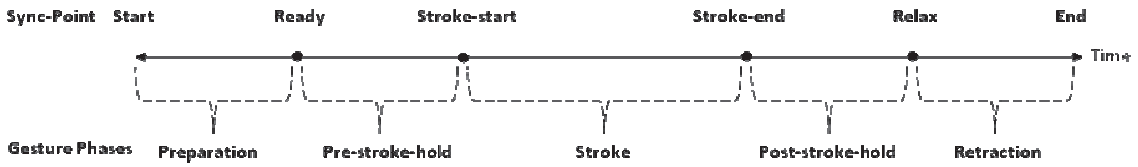


La vitesse "maximale" des gestes d'un robot physique (ex: Nao) est limitée par la vitesse des articulations. Il faut pré-estimer les durées minimales pour faire les gestes. Par contre, la vitesse "minimale" n'est pas encore étudiée dans cette thèse.

5.3. En ce qui concerne la troisième question de recherche "Comment synchroniser les gestes avec la parole?", on a utilisé le langage de représentation BML pour spécifier les comportements multimodaux avec contraintes à réaliser (Kopp et al., 2005).

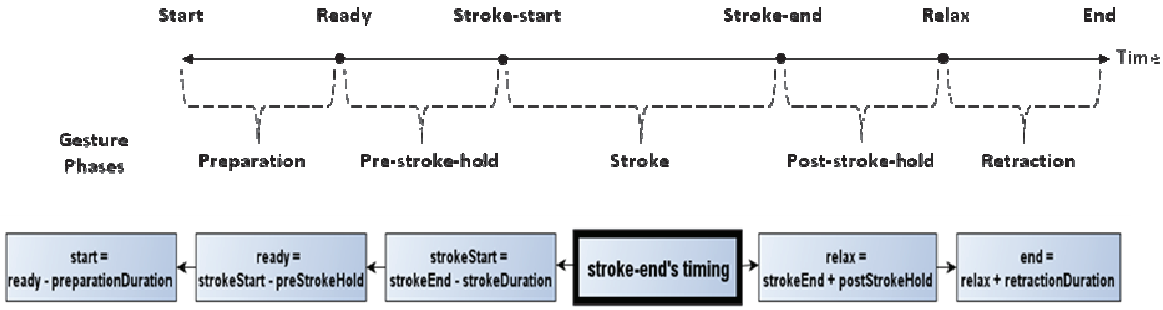
```
<bml>
<speech id="s1" start="0.0" language="english" voice="acapela">
<description level="1" type="gretabml">
<reference>tmp/from-fml-apml2.pho</reference>
</description>
<tm id='tm1'/> I
<tm id='tm2'/> am
<tm id='tm3'/> sad
...
</speech>
    <gesture id="sadness"  start="s1:tm2" end="s1:tm3" stroke="0.3">
        <description level="1" type="gretabml">
        <reference>emotion=sadness</reference>
        </description>
    </gesture>
</bml>
```

Ce langage nous permet de représenter la forme des signaux et la relation temporelle entre eux. La synchronisation des gestes avec la parole est fait par une adaptation de la performance des gestes à la structure du discours. En détail, l'information temporelle des gestes dans les balises BML sont relative à la parole par des marqueurs de synchronisation temporelle. Le temps de chaque phase gestuelle est indiquée par des points de synchronisation. Ces points divisent un geste un des phases dans lesquelles la phase de stroke (apogée) coïncide ou précède la parole (McNeill, 1992).

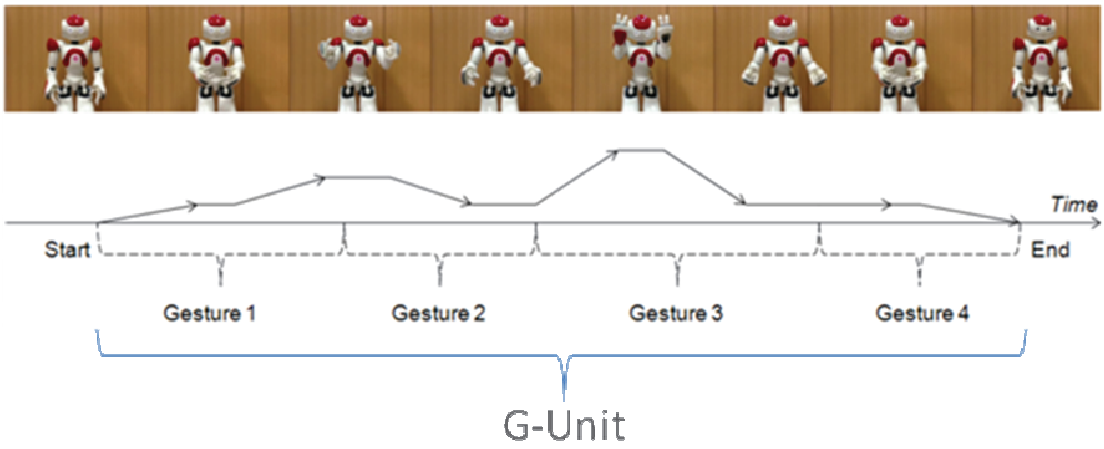


On a définit trois étapes pour planifier les gestes: 1) Calculer les durées des phases (préparation, stroke, rétraction) d'un geste en utilisant la loi de Fitts; 2) Calculer la valeur réel des syns points d'un geste. Les valeurs de la phase de stroke sont calculées à partir de la parole et les autres phases sont calculés à partir de la phase de stroke ; 3) Calculer les trajectoires gestuelles pour des séquences de gestes: on a basé sur une étude de Kendon (2004): Une trajectoire ou une unité gestuelle est un ensemble de gestes produits continuellement avec la parole sans phases de relaxation; les gestes sont co-articulés entre eux. Plus détaillé, une phrase gestuelle est composée de différentes phases (mais une seule stroke). Une unité gestuelle est composée de différentes phrases.

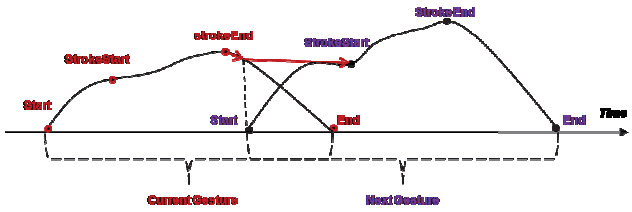
```
<bml>
    <speech id="s1" start="1"> <text>I don't <sync id="tm1"/> think so! </text>
    </speech>
    <gesture id = "g1" strokeEnd = "s1:tm1" mode = "BOTH_HAND" lexeme="DENY"\>
</bml>
```



strokeEnd ··· getTimeSynthesis(speech.tm1)
strokeStart = strokeEnd – strokeDuration
ready – strokeStart ·· preStrokeHold
start – ready – preparationDuration
relax = strokeEnd + postStrokeHold
end – relax + retractionDuration



5.4. La coarticulation des gestes existe entre deux gestes consécutifs quand le geste suivant commence lorsque le geste actuel ne finit pas et la phase de stroke du geste actuel a fini avant que la phase de stroke du geste suivant commence. Dans ce cas, il n'y a pas de phase de relaxation pour le geste actuel; le geste actuel faire une coarticulation avec le geste suivant.



5.5. Pour la cinquième question de recherche "Comment les rendre expressifs?", je vous rappelle que l'expressivité est définie comme la manière avec laquelle un geste est exécuté (Poggi et Pelachaud, 2008). Cette manière est concrétisée par un ensemble de paramètres des dimensions gestuelles (Hartmann et Pelachaud, 2006).

- L'amplitude des mouvements (SPC)
- La vitesse des mouvements (TMP)
- La puissance d'exécution (PWR)
- La fluidité des mouvements (FLD)
- La répétition de l'apogée du geste (REP)

Jusqu'à maintenant, on a implémenté trois paramètres pour le robot Nao. Ils sont:

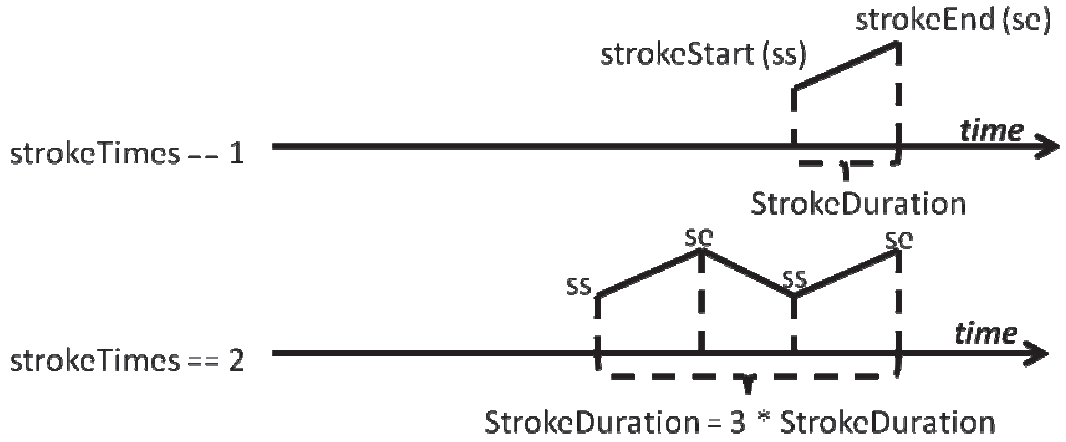
1) l'extension spatiale SPC change l'amplitude des mouvement (ex., élargi vs. contracté) pour tous les trois dimensions (vertical, horizontale, frontale). Cependant, certaines dimensions sont fixées afin de maintenir la signification du geste. Le paramètre SPC est appliqué uniquement aux dimensions disponibles. Par exemple, pour le geste d'arrêt, seul la dimension frontale est modifiable.

```
<gesture lexeme="stop" mode="RIGHT_HAND">
    <phase type="STROKE">
        <hand distanceFixed="false"
              horizontalFixed="true"
              verticalFixed="false">
            <verticalLocation>YUpperC</verticalLocation>
            <horizontalLocation>XC</horizontalLocation>
            <locationDistance>ZMiddle</locationDistance>
            <handShape>form_open</handShape>
            <palmOrientation>AWAY</palmOrientation>
            <fingersOrientation>UP</fingersOrientation>
```

2) l'extension spatiale TMP change la durée du mouvement (ex., rapide vs. lente). On a utilisé la durée calculée par la loi de Fitts et la durée minimale pour calculer ce paramètre.

3) Pour la répétition REP, le système décide combien il y a de répétitions de la phase de stroke en tenant compte de la valeur du paramètre REP et du temps disponible. Dans tous cas, la

synchronisation devrait être maintenue. C'est à dire que le sync point stroke-end coïncide ou précède les mots accentués.



6. Architecture du système

L'approche proposée dans cette thèse est de s'appuyer sur le système d'agent conversationnel animé Greta pour contrôler des comportements des agents. Le système GRETA suit l'architecture de SAIBA (Figure 1). Il se compose de trois modules séparés: le premier module, la planification des intentions, définit les intentions communicatives que l'agent vise à transmettre. Le deuxième module, la planification des comportements, planifie les comportements correspondants à réaliser. Le troisième module, la réalisation des comportements, réalise les comportements planifiés. Le résultat du premier module est l'entrée du deuxième module via une interface décrite avec le langage de représentation FML, Function Markup Language [2]. La sortie du deuxième module est encodée avec un autre langage de représentation BML, Behavior Markup Language [3], puis envoyée au troisième module. Les deux langages FML et BML sont représentés sous forme de XML et ne font pas référence aux paramètres d'animation spécifique de l'agent (e.g. articulation du poignet).

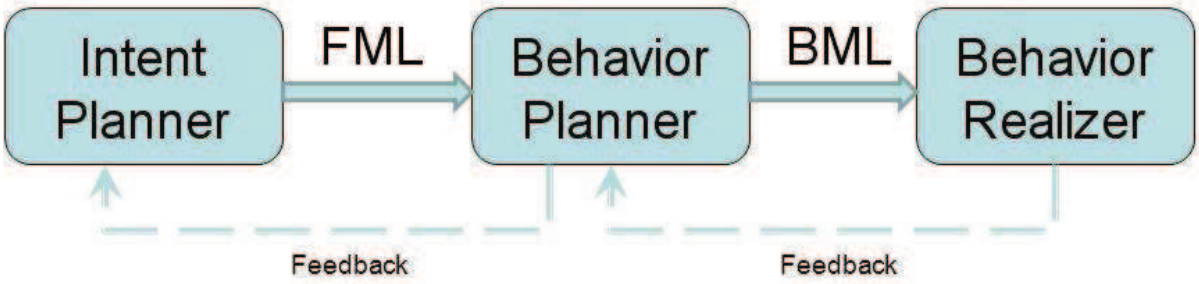


Figure 4: Architecture de SAIBA [3]

Nous voulons être en mesure d'utiliser le même système pour contrôler les deux agents (i.e. l'agent corporel virtuel Greta et l'agent physique Nao). Cependant, le robot et l'agent n'ont pas les mêmes capacités (par exemple, le robot peut bouger ses jambes et le torse, mais n'a pas d'expression du visage et a des mouvements de bras très limités; tandis que l'agent virtuel n'a pas la notion de la gravité). Pour cette raison, les comportements non-verbaux du robot ne peuvent pas être toujours identiques à ceux de l'agent virtuel. Par exemple, le robot n'a que deux configurations de la main, ouverte ou fermée, il ne peut pas étendre un seul doigt. Par conséquence, pour faire un geste déictique il doit étendre tout son bras vers une cible plutôt que d'utiliser un index tendu comme le fait l'agent virtuel. Pour contrôler les comportements communicatifs du robot humanoïde et ceux de l'agent virtuel, tout en tenant compte de leur contrainte physique, nous considérons deux lexicons (i.e. les dictionnaires des comportements non-verbaux), un pour le robot et l'autre pour l'agent. Du même fichier BML émis par la planification des comportements, on instancie les balises de BML de l'un ou l'autre lexicon (cf Figure 2). Les autres parties du système GRETA restent les mêmes. Etant donné un ensemble d'intentions et d'émotions à transmettre, le système GRETA calcule, grâce à la planification des comportements, la séquence correspondante de comportements spécifiés avec BML.

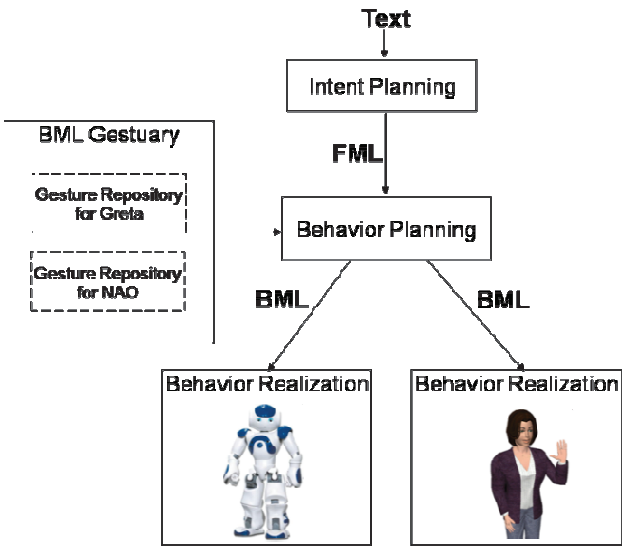


Figure 2: Une vue globale du système proposé

7. Lexicon

Dans le système GRETA, un lexicon est défini comme un dictionnaire de signaux multimodaux (e.g. la main, la tête, le regard, etc) que l'agent peut sélectionner, combiner et réaliser pour transmettre une intention communicative donnée. Chaque élément du lexicon a deux paramètres principaux: le nom d'une intention et un ensemble de signaux multimodaux correspondants [Mancini et al, 2008]. L'exemple suivant montre un ensemble de

comportements que l'agent utilise lorsqu'il veut communiquer la tristesse. Une contrainte est définie dans le tag *core* pour indiquer que l'agent doit utiliser l'expression triste du visage :

```
<behaviorset name="emotion-sadness">
<signals>
<signal id="1" name="down" modality="head"/>
<signal id="2" name="down" modality="gaze"/>
<signal id="3" name="sadness" modality="face"/>
</signals>
<core>
<item id="3"/>
</core>
</behaviorset>
```

Tous les signaux multimodaux, qui sont associés à une intention communicative dans le lexicon, sont définis dans les répertoires externes. Chaque agent (virtuel or physique) peut être caractérisé par son propre répertoire qui contient une description des signaux spécifiques. Par exemple, nous pouvons définir un agent qui montre des expressions asymétriques du visage ou celui qui n'a pas de gestes avec l'orientation du poignet vers l'intérieur. Le deuxième exemple est particulièrement important lors de l'élaboration d'un geste pour le robot Nao en raison de ses limites physiques.

7.1. Elaboration des lexicons

Un lexicon propre est élaboré pour le robot NAO, ainsi que pour l'agent virtuel Greta.

Pour s'assurer qu'il n'y ait pas de contradiction dans la transmission d'un ensemble donné d' intentions communicatives et d'états émotionnels, chaque lexicon doit avoir un élément transmettant un message similaire. Autrement dit, les deux lexicons contiennent les mêmes entrées pour les intentions. Par contre les signaux multimodaux associés aux intentions peuvent être différents. Le travail de Calbris (1983) sur les familles des gestes avec variantes est étudié pour élaborer ces lexicons.

- La taxonomie des gestes de Calbris (1983)
    - Les gestes dans une famille peuvent se différencier en forme, mais véhiculer un message similaire (e.g. la famille de la négation (Calbris, 1983, p.398))
    - Les paramètres de description d'un geste peuvent être divisés en deux parties, une partie invariante et l'autre variante. Les paramètres de la partie invariante doivent être consistants, tandis que les paramètres de la partie variante peuvent être ajustés sans changer la signification du geste. (e.g. La coupure en utilisant le tranchant de la main coupe (Calbris,

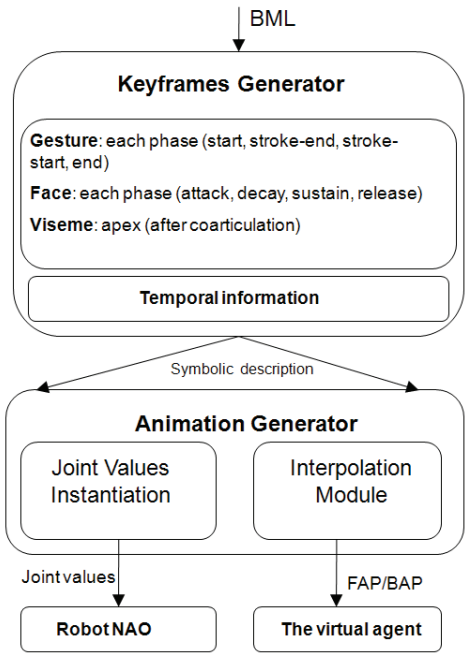
1983, p.498)). La coupure en utilisant la main coupe: On a plusieurs variantes mais l'aspect commun est le tranchant de la main coupe (la direction des mouvements est toujours suit la tranchant)

- Verticalement: coupure en deux, verticale division
- Transversalement, paume vers le sol: coupure en deux, transversale coupure totale, définitive
- Transversalement, paume vers le ciel: coupure à ras du sol couper l'herbe sous le pied saper quelqu'un ou quelque chose

Dans le cardre du projet Gvlex les prototypes gestuels dans les répertoires des gestes sont élaborés à partir d'informations annotés du corpus de vidéos ContTact par Martin et al [20] . Dans ces vidéos, 6 sujets humains racontent une même histoire appelée « Trois petits morceaux de la nuit ». Ils ont été enregistrés avec deux caméras, une de face et une de profil. Les gestes expressifs et les profils individuels sont annotés avec un logiciel d'annotation de vidéos (i.e. Anvil).

8. Behavior Realizer

Le tâche principale du Behavior Realizer (BR) est de générer l'animation de l'agent (virtuel ou physique) à partir d'un message BML. Ce message contient les descriptions des signaux et leur information temporelle à réaliser. Le processus est divisé en deux étapes principales: la première étape, appelée Keyframes Generator (KG) peut être utilisée en commun pour les deux agents tandis que la seconde, Animation Generator (AG) est spécifique à un agent donné. La figure 3 présente la structure de notre Behavior Realizer. Dans les sous-sections suivantes, je présente ces modules en détail.



8.1. Keyframes Generator

Dans cette étape, les signaux décrits symboliques dans un message BML sont instanciés. Ils peuvent être des expressions faciales, du regard, des mouvements de la tête, du torse ou des gestes de la main. Dans le cas particulier du robot, le message BML ne contient que les signaux des gestes de la tête et de la main. Le Keyframes Generator synchronise les comportements non-verbaux avec la parole. Dans notre système, la synchronisation entre des signaux multimodaux est réalisée par l'adaptation des signaux non-verbaux à la structure du discours. Cela signifie que l'information temporelle des comportements non-verbaux dans les balises BML sont relatives à la parole; ils sont spécifiés par des marqueurs des temps (i.e. *time markers*). Dans le cas des gestes, l'information temporelle de chaque comportement corespond aux phases gestuelles. Comme illustré dans la Figure 4, ils sont encodés par sept points de synchronisation: *start*, *ready, stroke-start, stroke, stroke-end, relax* et *end*. Ils divisent un geste en plusieurs phases de réalisation, dans lequel la partie la plus significative se produit entre *stroke-start* et *stroke-end* (i.e. la phase d'apogée ou *stroke*). La phase préparatoire arrive de *start* à *ready*. Cette phase met les articulations corporelles (e.g. la main et le poignet) à la position où aura lieu le *stroke*. Selon des observations de McNeill (1992), la phase de stroke coïncide ou précède la parole. Dans notre système, la synchronisation entre les gestes et la parole est assurée en calculant le temps de démarrage de la phase *stroke* pour qu'elle coïncide avec les syllabes accentuées. Donc, le système doit estimer le temps, t_pre, requis pour la réalisation de la phase préparatoire afin de s'assurer que le *stroke* soit réalisé avec les syllabes accentués. Cette estimation est faite en calculant la distance entre la position actuelle de la main et la position prochaine souhaitée et en calculant le temps qu'il faut pour effectuer la trajectoire gestuelle (t_traj). Dans le cas où le temps disponible ne suffit pas pour faire la phase préparatoire (t_pre<t_traj), tout le geste est annulé, ce qui laisse du temps libre pour préparer le geste suivant.

Le résultat de Keyframes Generator est un ensemble de keyframes. Chaque keyframe contient une description symbolique de chaque phase d'un geste (*start, stroke-start, stroke-end, end*).

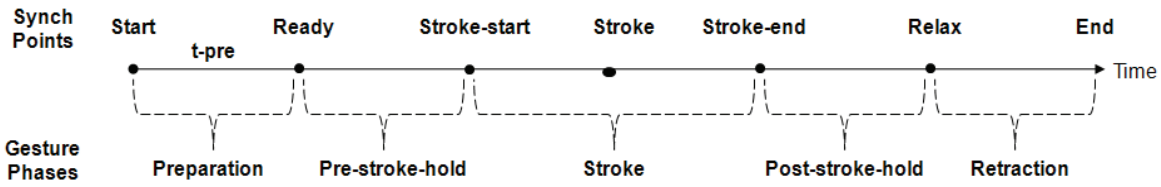


Figure 4. Les points de synchronisation du geste (SAIBA)

8.2. Animation Generator

Pour calculer l'animation à partir des keyframes, nous avons besoin d'utiliser un module spécifique pour chaque agent. Tandis que le module Keyframes Generator est commun à tous les agents, les calculs d'Animation Generator sont dépendants de chaque agent. Le module reçoit les keyframes du processus précédent et calcule les valeurs des paramètres de l'animation. Pour l'agent virtuel Greta, nous utilisons un module d'interpolation (i.e. *Interpolation Module)* et pour le robot Nao nous utilisons un module d'instanciation des valeurs d'articulation (*Joint Values Instantiation Module*) (voir Figure 3).

9. Implémentation

Le premier résultat obtenu est que le robot est contrôlé en utilisant le système développé. A partir des intentions sous format d'un message FML, GRETA planifie des gestes et ensuite retourne les keyframes correspondantes à l'animation. Chaque keyframe contient l'information temporelle et les informations gestuelles du robot telles que la forme de la main, la position et la direction du poignet. J'ai développé un module de Joint Values Instantiation (voir Figure 3) qui reçoit ces keyframes et les traduit en valeurs d'articulation du robot. L'information temporelle et les valeurs d'articulation sont envoyées au robot. Grâce à ces informations, l'animation est obtenue à l'aide d'un mécanisme d'interpolation disponible dans le robot.

Deux paramètres d'expressivité gestuelle de Greta sont implémentés pour Nao: 1) l'extension spatiale (SPC) pour changer l'amplitude des mouvements (e.g. large vs. étroit) et 2) l'extension temporelle pour changer la durée des mouvements (e.g. rapide vs. lent). Ces modulations du mouvement sont faites lors de la transformation des gestes symboliques en valeurs d'articulation du robot.

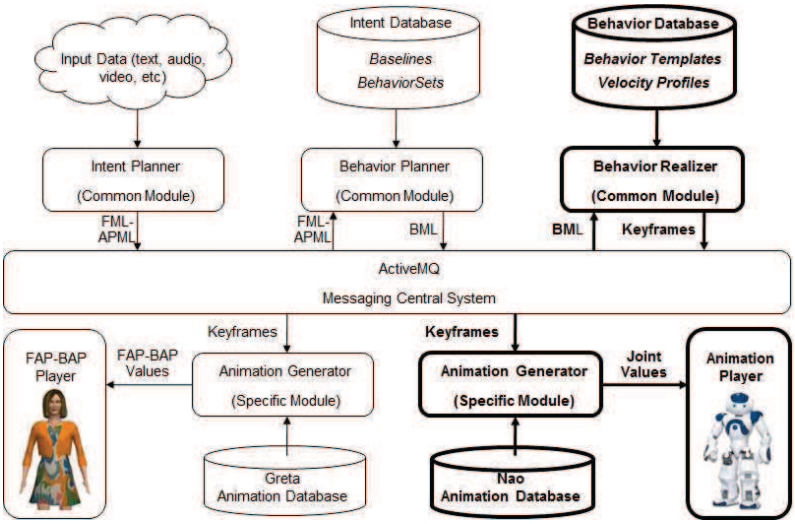


Fig 6. L'architecture du système

10. Evaluation

Afin de valider notre modèle de gestes expressive, nous avons mené une expérience perceptive. Nous voulions évaluer comment l'expressivité et le temps des gestes ont été perçus par les utilisateurs. Grâce à cette expérience, nous avons évalué la qualité de l'expressivité du geste pour les dimensions de l'extension spatiale et de l'extension temporelle ainsi la qualité de la synchronisation entre les gestes et la parole. Les résultats de cette expérience pourraient être utilisés non seulement pour valider notre modèle de gestes expressifs pour un robot humanoïde, mais aussi de répondre à la question de recherche: "Que ce soit un robot physique peut réaliser des gestes avec expressivité?".

Soixante-trois participants (27 femmes et 36 hommes) ont participé et répondu aux questionnaires de notre expérience. L'âge des participants variait entre 23 et 67 ans (moyenne = 37,02, écart = 12,14). Tous les participants étaient francophones de l'Ecole Nationale Supérieure des Télécommunications invités grâce à nos emails d'invitation.

Après avoir regardé 9 morceaux de vidéos du robot Nao via les interfaces Web, 48 participants (76%) ont accepté que les gestes du robot ont été synchronisés avec la parole, dans lequel 23 participants (36%) ont donné un accord léger et 25 participants (40%) ont donné un accord ou un accord solide. En ce qui concerne l'expressivité du geste, 44 participants (70%) ont convenu que les gestes du robot étaient expressive dont 24 participants (38%) ont donné un accord léger et 20 participants (32%) ont donné un accord ou d'accord fort.

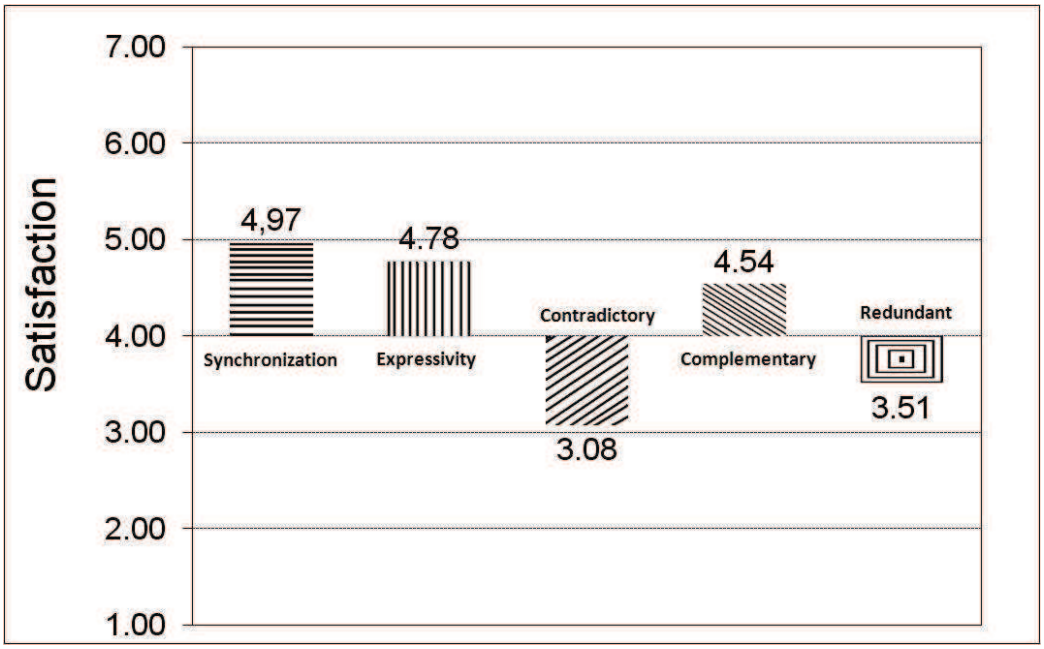


Fig 7. La satisfaction de l'évaluation

Lorsque nous avons élaboré les gestes pour le robot Nao, la forme des gestes était basée sur les gestes des acteurs réels . Notre modèle de gestes expressifs devrait reproduire des gestes humains pour le robot dans une manière que leur signification et leur forme ne sont pas modifiées par rapport des gestes originaux des acteurs. Par conséquent, si les gestes faits par les acteurs étaient complémentaires (redondants ou contradictoires respectivement) avec la parole, les gestes du robot devraient transmettre les informations similaires. Toutefois, cette conclusion doit être vérifiée par une analyse de la complémentarité (de la redondance ou du contradiction respectivement) des gestes et la parole des acteurs. Parce que le robot Nao a des contraintes physiques, certains gestes ne pouvaient pas être complètement reproduit à partir des gestes humains. Par exemple, le geste de l'emblème français "Je ne sais pas" est constitué d'un haussement d'épaules et d'un mouvement de bras avec la paume ouverte en face de l'interlocuteur. Cependant, le robot peut exécuter les mouvements des mains seulement. Pas de haussement d'épaules peut être modélisé. D'un même problème, dans le geste de l'emblème français "J'ai faim", les deux mains sont mises sur le ventre, mais le robot ne peut pas faire correctement d'une telle manière. Lorsque le robot fait une présentation gestuelle incomplète, il ne donne pas d'assez d'informations pour les participants. En outre, nous n'avons pas un outil flexible pour élaborer précisément des gestes pour le robot. Ainsi, la forme de certains gestes perdent une partie de leur iconicité et donc devient incompréhensible. Par ailleurs, selon Habets et al. (2011), la coordination temporelle entre les gestes et la parole est aussi un facteur importante sur l'interprétation des gestes. Dans le cas de l'absence de la parole, un geste peut être interprété par de multiples significations par les participants. L'interprétation conforme du geste doit être soutenu par son discours qui l'accompagne. Lorsque deux modalités ne sont pas correctement synchronisés, le geste peut être compris par une phrase incompatible (par exemple, par mots plus tard ou plus tôt).

En résume, les résultats principaux de l'expérience perceptive du robot Nao sont: (1) les participants de l'expérience ont apprécié que les gestes du robot ont été synchronisée avec la parole; (2) la majorité des participants ont estimé les gestes du robot comme étant expressive, et (3) la majorité des participants ont trouvé que les gestes du robot et la parole ont transmis des informations complémentaires. Ces résultats expérimentaux ont validé que notre modèle a généré les gestes expressifs et synchronisé avec la parole. Cependant, les résultats ont montré une limitation importante de notre système. Le naturalité des gestes du robot n'a pas été jugé acceptable dans cette expérience. La plupart des participants n'étaient pas d'accord que les gestes du robot étaient naturels. Ce problème vient en partie de certaines contraintes physiques d'un robot réel et en partie de notre algorithme qui n'a pas complètement reconstruire les gestes naturels, comme la forme et la vitesse. En particulier, notre système n'a pas encore mis en œuvre certaines dimensions d'expressivité du geste comme la puissance, la tension et la fluidité de gestes. En outre, la différence de vitesse entre les phases d'un geste (i.e., préparation, stroke, rétraction) n'a pas encore étudié dans ce modèle.

11. Conclusion

Le rapport présente un modèle de génération des gestes expressifs communicatifs accompagnant la parole pour un agent humanoïde. Le modèle est conçu d'une telle manière que la plupart de ses processus est indépendant à la représentation des agents. Jusqu'à maintenant, le modèle est utilisé pour contrôler l'agent virtuel Greta et le robot physique humanoïde Nao.

Le modèle est intégré dans la plateforme de génération de comportements multimodaux GRETA. Tandis que l'étape de sélection des gestes a déjà été implémenté dans la plateforme, mes travaux de la thèse se focalise sur l'étape de réalisation des gestes. C'est à dire que mes modules développés planifient et instancient les gestes sélectionnés par une action concrète des mouvements de la main et la jouer par l'agent. Les travaux de recherche sont liés à une étude des gestes humains accompagnant la parole et les résultats sont appliqués pour générer les gestes pour un agent humanoïde. Il y a trois questions de recherche à adresser dans ces travaux. Premièrement, des gestes humains sont encodés et reproduits d'une telle sorte que ces gestes sont réalisables pour les agents. Un ensemble de propriétés d'un geste tels que la forme de la main, la position du poignet, la forme de la trajectoire, etc. est utilisé afin d'encoder les gestes. Deuxièmement, les gestes sont planifiés pour synchroniser avec la parole. Le modèle compte sur la relation entre les gestes et la parole pour calculer le temps des gestes. Troisièmement, ces gestes sont rendus expressifs. Dépendant sur la personnalité, l'état émotionnel actuel de l'agent, ses gestes sont variés en changeant les valeurs des paramètres d'expressivité. Les résultats de la recherche est appliqués aux deux modules de la plateforme GRETA: le premier module Behavior Realizer est commun pour tous les deux agents virtuel et physique et le deuxième module Nao Animation Geneorator est spécifique au robot humanoïde Nao.

Le module de Behavior Realizer travaille avec trois aspects de génération des gestes: la représentation des gestes, l'expressivité des gestes et la coordination entre les gestes et la parole.

En ce qui concerne la représentation des gestes, nous avons proposé un langage de spécification des gestes afin d'encoder les prototypes gestuels symboliquement dans le répertoire des gestes. Un prototype gestuel contient une seule la phase de stroke qui serait utilisé pour reproduire tout geste en temps réel. A cause des différences entre l'agent Greta et le robot Nao, chaque agent a son répertoire des gestes propre qui contient les gestes adaptés à sa spécification. Un répertoire des gestes est construit pour le robot Nao. L'élaboration de ce répertoire est basé sur des annotations gestuelles qui sont extraites des acteurs réels d'un corpus des vidéos de conteurs.

Concernant l'expressivité, le module utilise un ensemble des qualités gestuelles telles que l'extension spatiale, l'extension temporelle, la répétition de la phase de stroke afin de moduler la durée et la forme de la trajectoire des gestes.

Pour la coordination temporelle des gestes avec la parole, on a travaillé sur deux tâches: 1) une simulation de la vitesse des gestes humains et 2) la synchronisation entre les gestes avec la parole. La première tâche a été réalisée en utilisant la loi de Fitts afin de calculer la durée pour faire un mouvement de la main. Un robot peut demander plus temps pour faire un mouvement que le temps calculé avec la loi de Fitts. Dans ce cas, on a du pré-estimé la durée minimale pour faire un geste du robot afin de planifier les gestes correctement. La deuxième tâche de la synchronisation entre les gestes avec la parole est assurée par une adaptation des mouvements des gestes à l'exécution de la parole. Le temps des gestes est relative à la parole via des sync-points indiqués dans le langage de représentation BML.

L'implémentation de notre modèle a été validée par un ensemble des testes perceptifs sur le robot Nao. Nous avons voulu évaluer comment les gestes du robot sont perçus par des sujets humains sur le niveau de l'expressivité des gestes et sur le niveau de la synchronisation des gestes avec la parole ainsi que sur la fonction reliant le geste avec la parole. Les résultats ont montré que les gestes créés par notre modèle et animés par le robot Nao sont acceptables. 76% des participants trouvent que les gestes sont synchronisés avec la parole et 70% des participants trouvent les gestes expressifs.

INSTITUT
DES SYSTÈMES
INTELLIGENTS
ET DE ROBOTIQUE

www.isir.fr

Mohamed CHETOUANI
Mohamed.chetouani@upmc.fr
01 44 27 63 08
01 44 27 51 45

Paris, le 10 Juin 2013

Objet : Rapport en vue de la soutenance de la thèse de doctorat de Le Quoc Anh

Le Quoc Anh présente, en vue de l'obtention du Doctorat en Informatique délivré par TELECOM ParisTech, son manuscrit intitulé « Modèle de gestes expressifs pour un agent humanoïde ». Ce mémoire comprend 6 chapitres, plus une discussion. Le mémoire est bien rédigé, les articulations inter-chapitre sont limpides. Les figures et les tables sont soignées et bien commentées. La bibliographie est riche, avec de nombreuses références pertinentes.

Le chapitre 1, introductif, présente le cadre de l'étude, et une de ses motivations: le développement d'un modèle de gestes expressifs pour un humanoïde. Notons dans cette introduction des sections intéressantes permettant au lecteur de comprendre le positionnement scientifique riche (communication gestuelle, interaction humain-agent virtuel et évaluation perceptive), ce qui montre l'ouverture d'esprit scientifique du candidat.

Le chapitre 2 est consacré à la description des fondements théoriques des travaux réalisés à travers la présentation d'un ensemble modèles du geste. Le candidat précise également la nécessité d'intégrer la modalité parole, et ce à plusieurs niveaux de description, dans un modèle de génération du geste. Sur le plan de l'expressivité, une synthèse intéressante permet de dégager les dimensions importantes du geste. Une discussion intéressante sur les modèles de génération de gestes clos ce chapitre. On peut regretter une comparaison plus détaillée de ces modèles dans un esprit de positionnement scientifique. La conclusion de ce chapitre permet néanmoins de clairement comprendre les grands principes suivis pour la proposition du modèle de geste expressifs.

Le chapitre 3 regroupe les travaux de la communauté scientifique dans le domaine de la génération de gestes. Une distinction entre les interactions avec un agent virtuel et un robot humanoïde est expliquée. La description des modèles et méthodes permet de faire émerger des structures fonctionnelles avec plusieurs niveaux de planification : de l'intention à la réalisation. Ce chapitre propose un bilan comparatif des approches très judicieux. On peut noter une description pédagogique des enjeux théoriques et pratiques

des approches des modèles proposés dans la littérature. Un ensemble de questions scientifiques plus ou moins ouvertes sont avancées avec une section permettant de clairement comprendre la démarche suivie par le candidat. Les justifications sont pertinentes et montrent une volonté de proposition de méthodes innovantes pour l'interaction humain-agent humanoïde.

Le chapitre 4 décrit la démarche de conception dont l'objectif est de s'assurer de la fonctionnalité du modèle de génération de gestes. Sur la base du modèle GRETA, Le Quoc Anh précise la démarche de conception, les adaptations nécessaires, les processus de communication entre les différentes unités de planification et de réalisation du geste pour un robot humanoïde. Là aussi, la démarche est ambitieuse avec une prise en compte des spécificités de la robotique. Une section sur l'animation d'un agent humanoïde permet de préciser la problématique et les enjeux. Le modèle proposé par le candidat se veut générique afin d'animer à la fois des agents virtuels et des systèmes robotiques. Pour ce faire, un répertoire de gestes représentés sous forme d'actes de communication est proposé. Il s'agit d'une approche judicieuse permettant à un système de planification commun de gérer ces actes durant l'interaction. L'animation effective des gestes est séparée de la planification des différentes phases d'un geste donné. A noter que la multimodalité (visage et parole) est très judicieusement intégrer à la phase de réalisation du geste. Le candidat pointe les éléments clefs de l'animation générique d'agents virtuels et robotiques à savoir la gestion de l'espace de travail, surtout celui du robot, et la dynamique temporelle. Cette dernière doit être nécessairement adaptée aux spécificités du robot. On peut regretter des liens vers les problématiques de singularités et/ou de stabilités des gestes dans l'espace de travail d'un robot. Une section très intéressante sur la proposition d'un modèle de génération de gestes expressifs clos ce chapitre. La qualité de cette section montre le recul du candidat sur la thématique. De plus, elle s'avère fort utile à la structuration des différents chapitres suivants.

Le chapitre 5 représente le cœur de la thèse et traite de la question de l'implémentation du modèle de génération de gestes. Une description très précise du répertoire de gestes est proposée. Ce chapitre regroupe, pour une grande partie, l'analyse de la littérature réalisée par le candidat. Le modèle se nourrit de cette analyse. Une proposition importante de la thèse réside dans la définition d'une liste d'indices du geste. Ces indices sont nécessaires et sont tout à fait pertinents pour la génération du geste. On retrouve très nettement l'ambition de développer des modèles utiles à la communication avec l'humain avec des indices cohérents et potentiellement interprétables par l'humain. A noter que ces indices permettent d'étendre la description générique de comportements (BML). Le niveau d'abstraction obtenu offre la possibilité d'envisager un modèle générique de génération de gestes expressifs. Dans une seconde partie, le candidat traitre de la problématique de la gestion du temps lors de la réalisation du geste. L'approche exploite la loi de Fitts généralement dans les études de contrôle moteur. Il s'agit d'une loi relativement simple mais d'une efficacité prouvée et, comme le précise le candidat, adaptée à des contraintes de temps réel. Cette loi n'est que très peu utilisée pour la génération de gestes communicatifs. L'estimation des paramètres de cette loi sur des gestes expressifs est réalisée via une analyse de gestes humains. La robotique impose des contraintes physiques, et comme indiqué par le candidat, sont dépendantes du robot utilisé. La solution mise en œuvre par le candidat consiste à définir un répertoire de gestes en intégrant la vitesse maximale de réalisation du geste par le robot. On peut regretter un modèle de planification exploitant automatiquement la configuration de l'espace de travail du robot. Cependant, il faut noter que cette dimension demande d'élargir le spectre des domaines traités déjà multidisciplinaire. La seconde partie de ce chapitre est dédiée à une description très précise des étapes de génération de gestes expressifs pour GRETA ainsi que pour le robot NAO. Cette seconde partie est très riche

et permet de comprendre les différentes étapes de traitement notamment la gestion de la dynamique temporelle : de la réalisation d'un geste donné à la planification de plusieurs gestes (incluant la co-articulation). Il faut noter ici le souci de détails mais également le recul du candidat dans la description du modèle de génération. Une discussion clos ce chapitre mettant en évidence les caractéristiques du modèle et ouvrant la voie vers une série d'analyses ainsi que des futures améliorations.

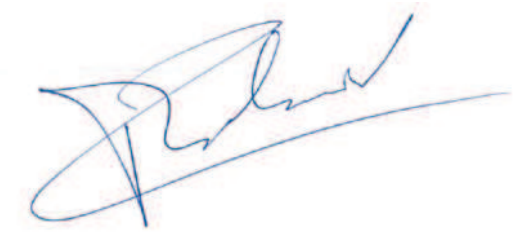
Le chapitre 6 a pour vocation l'évaluation du modèle de génération de gestes expressifs, il s'agit d'évaluation perceptive visant à mesurer la qualité des gestes. Les expériences sont menées dans un contexte réaliste, avec des hypothèses claires et pertinentes. Là aussi, les analyses sont menées de manière rigoureuse en combinant analyses qualitative et quantitative. L'approche proposée considère quatre situations interactives : synchronisation geste-parole, désynchonisation geste-parole, synchronisation geste-parole dans un modèle expressif et synchronisation parole-geste dans un état neutre du modèle d'expressivité. Plusieurs analyses statistiques permettent de faire dégager des résultats intéressants. Une discussion finale permet de relever les points essentiels de l'étude, il faut noter que le candidat est très conscient des limites et ne cherche pas à sur-interpréter les données, notamment sur la perception du caractère naturel ou pas des gestes.

Le chapitre 7 résume et discute les principales contributions de la thèse. La discussion rappelle les points forts de la thèse en robotique interactive, et dresse des perspectives très prometteuses, notamment vers la planification temporelle ou bien encore la généralisation de l'approche à d'autres robots. Le candidat identifie les limites de l'approche et propose un grand nombre d'axes de recherches tout à fait pertinents. On notera tout particulièrement le recul scientifique de Le Quoc Anh qui se traduit par une analyse précise des limitations des systèmes proposés ou bien encore des difficultés rencontrées lors de l'évaluation

La maîtrise de Le Quoc Anh en interaction sociale, démontrée dans son manuscrit, devrait lui permettre à moyen terme d'être un acteur de sa communauté.

En conclusion, il s'agit d'un travail de qualité, original et novateur, reposant sur des hypothèses et des méthodologies nouvelles auxquelles le candidat a fortement et directement contribué, qui constitue un modèle de recherche pluridisciplinaire en interaction sociale (communication non-verbale, agent virtuel, robotique humanoïde, évaluation…). Je donne donc un avis très favorable à sa soutenance. La thèse pouvant être soutenue en l'état.

Mohamed CHETOUANI
Maître de conférences
Habilité à diriger des recherches



Institut des Systèmes Intelligents et de Robotique
UPMC – CNRS / UMR 7222
Pyramide Tour 55 - Boîte courrier 173
4 Place Jussieu – 75252 PARIS cedex 05 – France
Tél. +33 (0)1 44 27 51 31 / 51 41 – Fax +33 (0)144 27 51 45

Sous la co-tutelle de

LAAS-CNRS

Laboratoire d'Analyse et
d'Architecture des Systèmes

7, avenue du Colonel Roche
31077 Toulouse Cedex 4

T. +33 (0)5 61 33 62 00
F. +33 (0)5 61 55 35 77
www.laas.fr

Rapport sur le mémoire de thèse de Le Quo Anh

Modèles de gestes expressifs pour un agent humanoïde

Rachid ALAMI

Directeur de Recherche CNRS

Le travail de thèse de Le Quo Anh se situe dans le cadre général de l'interaction homme-machine et traite plus précisément des questions liées à la synthèse de gestes expressifs par des agents d'aspect humanoïde.

La contribution de Le Quo Anh a porté plus particulièrement sur l'étude de modèles suffisamment riches et génériques pour, d'une part, couvrir une large classe de mouvements et prendre en compte explicitement les interactions entre le geste et la parole et, d'autre part, pour être effectivement applicables sur des agents aussi bien virtuels, tels que les agents conversationnels, que sur des robots physiques de forme humanoïde. Il s'agit là d'un sujet très intéressant et traité ici avec la volonté d'avancer à la fois sur la modélisation et sur la conception et implantation d'un système qui synthétise et réalise des gestes expressifs. Enfin, notons un souci permanent de pertinence et d'évaluation des performances des fonctions développées.

Le manuscrit comprend six chapitres et une conclusion.

Le chapitre 1, introductif, pose le contexte et les motivations. Il définit le problème et l'ambition de construire une plateforme commune pour différents types d'agents, qu'ils soient virtuels ou réels. Enfin, Le Quo Anh résume les contributions de sa thèse.

Le chapitre 2 est consacré aux aspects fondamentaux portant sur la compréhension du geste humain et qui servent de base à l'élaboration des modèles de génération de geste. L'auteur présente une classification des gestes issue de plusieurs études ainsi que des éléments de littérature récente permettant de bien comprendre la structure des gestes (en phases), le couplage avec la parole et enfin les éléments clefs d'analyse de l'expressivité. Sur cette base, deux modèles théoriques de génération de gestes sont présentés et discutés. Ce chapitre est bien construit et très instructif.

Le chapitre 3 porte sur une analyse de l'état de l'art des systèmes de génération de gestes de communication aussi bien pour des agents virtuels, bien développés, que pour des initiatives de réalisation sur des robots physiques. Les différents systèmes

LAAS-CNRS
Laboratoire d'Analyse et d'Architecture des Systèmes

7, avenue du Colonel Roche
31077 Toulouse Cedex 4
T. +33 (0)5 61 33 62 00
F. +33 (0)5 61 55 35 77
www.laas.fr

sont analysés et comparés selon plusieurs aspects. L'auteur a notamment produit des tableaux synthétiques, très intéressants qui permettent à la fois d'avoir une vue globale de l'ensemble mais aussi de situer ses propres contributions dans le paysage.

Le chapitre 4 est consacré à la conception du système proposé par M. Le Quo Anh. dans une première partie il décrit le système GRETA pré-existant au travail et comment il a été adapté pour à la fois permettre de piloter différents systèmes aussi bien virtuels que robotiques. Le choix, très pertinent, a consisté à limiter la spécificité au niveau du générateur de l'animation. Les autres composants (*Intent Planner*, *Behavior Planner* et *Behavior Realizer*) sont communs. Dans une deuxième partie l'auteur explicite à quel niveau le modèle traitant de l''expressivité est mis en oeuvre: au niveau générique, dans le *Behavior Realizer* et au niveau spécifique à l'agent, dans le composant appelé *Animation Realizer*.

Le chapitre 5 traite de l'implantation du système et plus précisément des composants qu'il a développé: un *Behavior Realizer* générique et un *Animation Realizer* spécifique au robot Nao. Le *Behavior Realizer* proposé est capable de traiter les "comportements" traduisant des gestes expressifs. Ceci a été possible grâce à une extension du langage BML qui permet de spécifier des gestes expressifs selon les aspects mentionnés dans le chapitre 2 et 3.

Des expérimentations on été conduites pour valider le système. Elles sont décrites dans le chapitre 6. Elles ont consisté à évaluer comment l'expressivité et les rythmes des gestes synthétisés par le système étaient perçu par des utilisateurs humains. Cette étude, sérieuse et poussée, a été réalisée en utilisant le robot Nao. Elle est très intéressante car elle montre à la fois une pleine pertinence du système sur les aspects essentiels de la synchronisation du geste et de la parole et sur la capacité du système de programmer des gestes expressifs. Elle montre des limitations sur le caractère "naturel" des mouvements du robot. Ceci est du à la fois aux limitations intrinsèques du robot mais aussi soulève la nécessité d'enrichir encore les modèles notamment sur les aspects dynamique et fluidité du mouvement.

Enfin, la conclusion (chapitre 7) résume les contributions principales du travail et discute des perspectives à court et moyen terme.

Le manuscrit est très clair, avance de manière progressive vers un système complet et pertinent. La contribution est substantielle sur plusieurs aspects. Le Quo Anh fait preuve d'une bonne connaissance de l'état de l'art. En ce sens, le document est très instructif. En effet, les chapitres 2 et 3 incluent une analyse fine des contributions existantes.

LAAS-CNRS

Laboratoire d'Analyse et
d'Architecture des Systèmes

7, avenue du Colonel Roche
31077 Toulouse Cedex 4

T. +33 (0)5 61 33 62 00
F. +33 (0)5 61 55 35 77
www.laas.fr

Ce travail porte donc sur plusieurs aspects: la spécification du geste sur la base de "templates" abstraits, la prise en compte de l'expressivité du geste au niveau d'un processus de calcul en-ligne de l'animation. Enfin l'ordonnancement des différentes étapes des gestes prend en compte explicitement un couplage fort avec la production de la parole. Enfin, l'implantation a permis de montrer l'utilisation effective du système aussi bien sur l'agent virtuel Greta développé par l'équipe que sur le robot humanoïde Nao de la société Aldebaran.

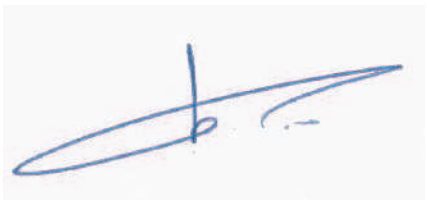
Les contributions portent sur 1) la construction d'un langage de représentation des gestes qui permet d'encoder de manière symboliques les gestes humains, 2) l'élaboration d'un répertoire de templates abstraits de gestes sur la base du langage proposé 3) l'animation des gestes aussi bien sur le robot physique que sur l'agent virtuel avec la prise en compte explicite des contraintes induites par la synthèse de la parole et par l'état émotionnel de l'agent en fin 4) l'évaluation de l'expressivité des gestes obtenus grâce à une sérieuse étude utilisateur.

Le système et les résultats sont intéressants et prometteurs. De plus, Le Quo Anh a aussi fait l'effort, très notable, d'évaluer son système puis de dégager et d'analyser les limites actuelles.

En conclusion, le travail décrit ici est de bonne facture. Il contribue de manière substantielle au domaine et ouvre la voie à des extensions diverses. Je donne donc un avis très favorable à la soutenance de thèse de Le Quo Anh.

Fait à Toulouse, le 14 juin 2013

Rachid ALAMI



cnrs

Contents

Chapter 1

Introduction

1.1 Objectives

The objective of this thesis is to develop an expressive gesture model for a humanoid agent. We define a humanoid agent as an intelligent autonomous machine with human-like shape including head, two arms, etc which are able to express certain human behaviors. It could be a screen agent or a physical agent. A screen agent developed by computer graphics is called an embodied conversational agent (ECA) or an intelligent virtual agent (IVA). A physical agent being developed by the robotic technology is called a humanoid robot or an anthropomorphic robot. Both of them are developed to improve the interaction between human users and a machine (i.e., a computer or a robot). In order to increase the believability and the life-likeness of such a humanoid agent in communication, it should be equipped with not only a human-like appearance but also similar human behaviors (Loyall and Bates, 1997). This thesis work focuses on the hand gesture communicative behavior.

Role of gestures

Communicative gestures are defined as a particular movement of hand-arms for the goal of communicating some meaning (Poggi and Pelachaud, 2008). They have an important role in communication (Goldin-Meadow, 1999; Kendon, 1994; Krauss and Hadar, 1999; McNeill, 1992). Gestures can be used to replace words

(e.g., emblems, pantomimes) (Hogrefe et al., 2011) or they accompany speech while bearing a relevant semantic contribution (Poggi, 2008). When accompanying speech, they convey information that may be complementary, supplementary or even contradictory to speech (Iverson, J. M., Goldin-Meadow, 1998; Poggi and Pelachaud, 2008). The expressivity of gesture helps the performer to attract the attention, to persuade the listeners and to indicate emotional states (Chafai N.E., 2007).

Problem definition

Such gestures can be produced by a human subject without any difficulties. However, it is a big challenge for a humanoid agent. There are two main questions to be answered in a gesture generation system: *Which gestures are selected to convey a given communicative intent?* and then *How to realize gestures naturally?*. My thesis deals with the later question (i.e., *How*). That means, given selected gestures, our system has to instantiate and schedule these gestures with a concrete action of hand-arm movements and then display them by an agent. This task relates to three main aspects of the human gestures reproduction: 1) define the form of gestures; for instance, the "Greeting" gesture is defined as an action of waving the open right hand being raised over the head; 2) model the expressivity of gestures; for instance, when the "Greeting" gesture is made in a happy state the hand moves quickly and energetically; 3) the temporal coordination of gestures and speech. For instance, the "Greeting" gesture has to happen at the same time with the word "Hello!" uttered by the gesturer.

Procedures

In this thesis some procedures were proposed to resolve the three main aspects of the gesture generation. For the form of gestures, a gesture lexicon containing symbolical gesture templates has been developed. A symbolical gesture template contains the significative description which is necessary to outline the gesture form for a concrete gesture. For instance, in the "Greeting" gesture, its template includes two hand configuration shifts which have the same shape "raised open hand" but different waving positions. The gesture expressivity is specified by a set of quality

parameters which define the manner to realize the gesture. For instance, the speed parameter defines how quick a gesture goes, or the power parameter defines how much energy a gesture carries, etc. The temporal synchronization of gestures and speech is ensured by adapting gesture movements to speech timing. For instance, a speech synthesizer integrated within our system generates speech from a given text and returns the duration for each synchronized word uttered by the gesturer. This speech timing is used to schedule gesture movements.

As a whole, the gesture model is integrated within a multimodal behavior generation system including an existing gesture selection module. After the gesture selection step, our model reproduces corresponding gestures from gesture templates. These gestures are planned to be expressive and synchronized with speech. After that, they are realized by a humanoid agent. To calculate their animation, gestures are transformed into key poses. Each key pose contains joint values of the agent and the timing of its movement. Then, an animation generator interpolates key poses to generate full animation for the agent.

Virtual agent vs. Robotic agent

A second objective of this thesis is that the expressive gesture model should be developed as a common gesture generation platform. It means that its processes can be used to control communicative expressive gestures for agents whose embodiments are different. This work was conducted within the framework of a French project, called GV-LEX (R. Gelin, C. d'Alessandro, O. Derroo, Q.A. Le, D. Doukhan, J.C. Martin, C. Pelachaud, A. Rilliard, 2010). The projects aims at equipping the humanoid robot NAO (Gouaillier et al., 2009) and the virtual agent Greta (Pelachaud, 2005) with the capability of producing gestures expressively while speaking.

Because the robot and the virtual agent have not the same behavior capacities (e.g., they have different degrees of freedom), they may not be able to display the same gestures but their selected gestures have to convey the same meaning (or at least similar meanings). For this reason, they should have two repertoires of gesture templates, one for the virtual agent and another one for the robot. These two repertoires have entries for the same list of communicative intentions. Given

an intent, the system selects appropriate gestures from either their repertoire. For instance to point at an object, Greta can select an index gesture with one finger. Nao has only two hand configurations, open and closed. It cannot extend one finger as the virtual agent does, but it can fully stretch its arm to point at the object. As a result, for the same intent of object pointing, while the Nao repertoire contains a gesture of whole stretched arm, the Greta repertoire contains an index gesture with one finger.

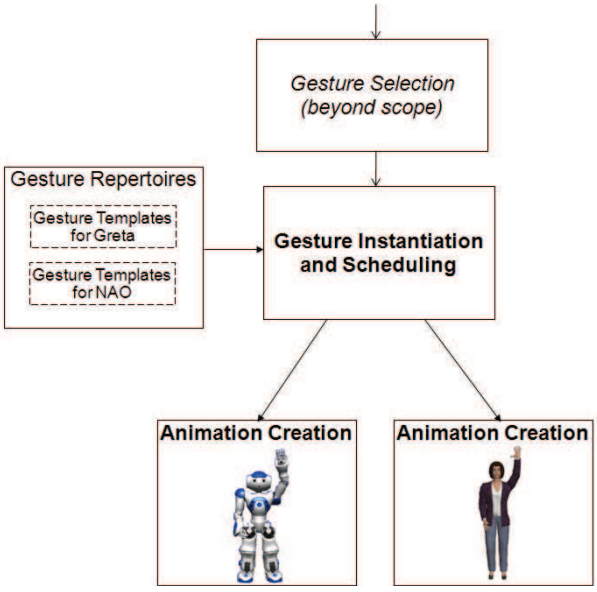


Figure 1.1: Proposed solution for two different agents

Additionally, the robot is a physical entity with a body mass and physical joints which have a limit in movement speed. This is not the case of the virtual agent. Our proposed solution is to use the same processes of reproducing and planing gestures, but different velocity parameters as well as different algorithms for creating the animation (i.e., MPEG-4 compatible animation parameters for Greta and joint-value parameters for Nao). An overview of this solution is illustrated in Figure 1.1.

In this research context, our contributions to the thesis were: 1) developing a gesture engine that encompasses the limitations of the robot's movement capabilities; 2) building a gesture representation language to encode human gestures symbolically; 3) elaborating a repertoire of gesture templates using our proposed

gesture representation language for the robot; 4) animating the mechanical robot hand-arms expressively in accordance with the speech and the emotional states; 5) evaluating robot expressive gestures via perceptive experiments. In addition to the contributions defined in the GVLEX project, this thesis considered the design of system able to produce equivalent movements on the robot and on the virtual agent.

Summary

In summary, the thesis work deals with an expressive gesture model which can control both virtual and physical agents. There are three features of this model. Firstly, the gesture specification (i.e. gesture form) is studied carefully when elaborating a repertoire of gesture templates. Secondly, the gesture expressivity is taken into account when gesture animation is computed on the fly from abstract gesture templates. Thirdly, the gestures are scheduled to ensure their execution are tightly tied to speech.

In this thesis, we present the first implementation of this model being used to control coverbal gestures of the Greta virtual agent and the Nao physical robot. This model was validated through a perceptive evaluation in a concrete case of the Nao storytelling robot.

1.2 Thesis Outline

The written thesis is divided into several chapters as illustrated in Figure 1.2.

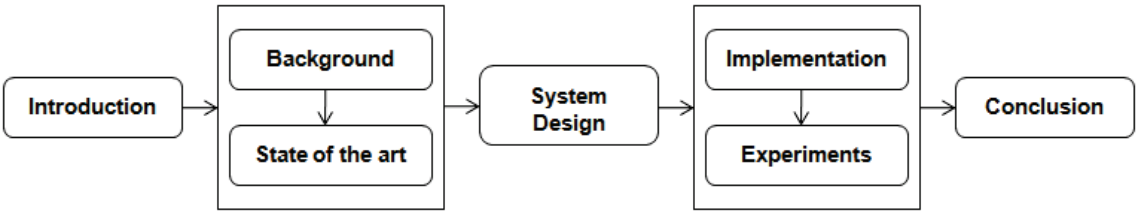


Figure 1.2: The structure of the thesis

Background This chapter presents some essential knowledge of human gestures that are necessary to build a gesture generation model. The chapter starts with a definition of expressive communicative gestures accompanying speech. Then we present, in the next sections, the classification, the hierarchy, the specification, the expressivity of hand-arm gestures. The last section introduces two featured theoretical gesture generation models which have inspired many implemented gesture models for virtual agents and humanoid robots.

State of the art This chapter is dedicated to present some state-of-the-art implemented gesture models as well as some other works related to our research. This chapter is divided into two parts: the first part resumes different approaches to create gestures for virtual agents; and the second part presents gesture engines for humanoid robots. The last section gives some differences between existing systems with our model.

System Design This chapter is divided into three sections. The first section presents an overview of an existing multimodal behavior generation system on which our gesture model is relying. The second section defines necessary requirements including database requirements and processing requirements to reach our objectives: having a general multimodal behaviors generation system to control different embodiments. In this section, a detailed analysis is given to show what are in common between various agents (being virtual or physique) and what are the differences between them generally. The third section presents how the existing system is redesigned to adapt to these requirements: the similar characteristics between agents can be put into common modules and the differences between them should be setup in external parameters (e.g., gesture repertoires) or separated modules.

Implementation This chapter presents the implementation of the expressive communicative gestures which are designed from previous chapter. There are three parts mentioned in this chapter. The first part presents a gesture representation language which is used to specify gesture templates in gesture repertoires. The second part presents a temporal coordination mechanism for synchronizing speech

and gestures. The third part presents how gesture expressivity is increased by implementing a set of predefined gesture dimensions for the gesture animation.

Experiments This chapter presents the procedures and the obtained results of our perceptive evaluations on the developed system. In the framework of the GV-LEX project, we built a gesture database using gesture annotations extracted from a video corpus of storytellers. The humanoid robot Nao, controlled by our system, reads a tale while doing hand-arm gestures. Then human users evaluate robot's gestures through precise questions such as the expressivity and the timing of gestures. The chapter finishes with an analysis of results obtained from the evaluations

Conclusion The thesis concludes with a short summary of our research which emphasizes on what we have achieved as new contributions along the thesis work with regard to the field of embodied conversational agents (both virtual and robotic). This thesis work is an attempt at developing a common multimodal behavior generation framework for different agents. Its results open new challenges to be solved in the future. Such remaining works are presented as the last words of the thesis.

Chapter 2

Theoretical Background

In this chapter, we present background information related to our gesture model for humanoid agents. It allows us to understand the mechanism of human gestures such as the structure of a gesture movement, the relation between gestures and speech as well as the meaning of a gesture signal, etc. This study is necessary to design a computational mechanism for a gesture model in which two main questions need to be answered: 1) how to encode a human gesture into a database so that the computer can interpret and afterwards reproduce its gesture without missing the signification; 2) how to simulate the co-articulation of a sequence of gestures naturally and plan them in accordance with other signals (e.g., speech). The answers for these questions should be based on the theoretical background that are described in the following sections.

With the first section, we give a definition of communicative expressive gestures. In the second section, a brief overview of the human hand gesture classification helps us to understand the signification of gestures. The structure of a human communicative gesture is studied in the third section. After that the relation of gestures with speech as well as the expressivity of gestures are presented in the next two sections. Then we present the gesture configuration in the following section. Lastly, two examples of theoretical gesture models are introduced and analyzed.

2.1 Definition of communicative expressive gestures

Human expressive gestures are defined as any body movements (e.g., a hand movement, a head movement, a torso movement, etc) that express one's feeling or thought (Kendon, 2004; OxfordDictionary, 2012; Poggi, 2008). In this thesis, we are only interested in hand gestures that occur during speech, and we focus on communicative expressive gestures. They are hand-arm gestures that transmit a certain information in communication (Butcher and Goldin-Meadow, 2000). Hence, they do not include gestures involved in direct object manipulation like touching or reaching something. Communicative gestures could be simple hand actions, used to designate, point at real or imaged objects. Or they could be complex hand-arm movements, accompanied with speech, to describe information in the speech (Kendon, 1994; McNeill, 1985). Following Poggi (2008), a communicative gesture is formed from a pair of (signal, meaning): 1) the signal is described by the shape and the movement of hands or arms; 2) the meaning represents a mental image or a propositional format in the mind of the gesturer that is conveyed through the signal.

The expressivity of gestures refers to the manner of gesture execution. Each individual may execute a gesture in different expressive ways (e.g., more or less intensely) depending on his personality or his current emotional states. The way we do gesture reveal who we are and how we feel. For instance, two persons are different from each others in the amplitude of their gestures. In another example, a person does gestures more slowly and less dynamically in a sad state. Expressive gestures reflect inner feeling states of the gesturer (Gallagher and Frith, 2004).

2.2 Gesture Classification

There are many classifications of hand gestures proposed by Efron (1941); Kendon (2004); McNeill (1992); Ekman and Friesen (1972); Poggi (2008). An overview of their gesture classification is resumed in Figure 2.1.

Following these researchers, hand gestures are firstly divided into two categories: inactive and active gestures. The inactive gestures refer to resting positions which are defined in (Dawson, 1967): "When immobilization is necessary,

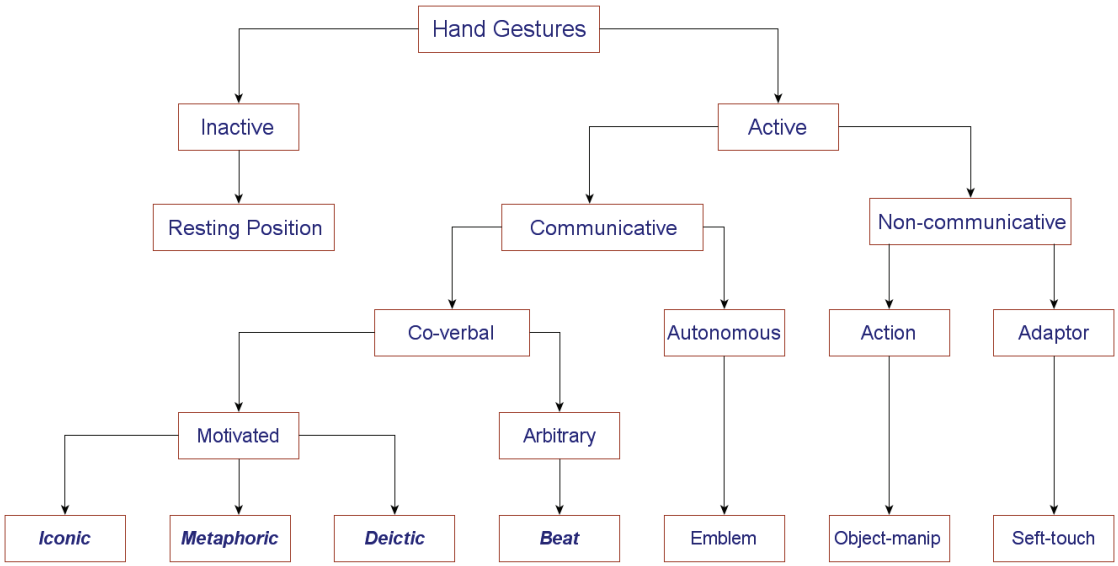


Figure 2.1: An overview of gesture classification is resumed from studies of Efron (1941); Kendon (2004); McNeill (1992); Ekman and Friesen (1972); Poggi (2008)

the hand should be placed in the *safe position* in order to limit the development of restriction of joint motion". Other gestures are part of the active gesture category. This category consists of certain gesture types including communicative and non-communicative gestures.

Communicative gestures are deictic, iconic, metaphoric and beat gestures. These gestures always accompany speech. Following Poggi (2002), iconic, metaphoric and deictic belong to a set of motivated gestures whose definition is: "A gesture is motivated when the meaning can be inferred from the signal even by someone who has never perceived it before; that is, when the meaning is linked to the signal in a non-random way". The iconic gestures illustrate a concrete event or object being said in speech (McNeill, 1992). For instance, the fingers of two hands form a circle when the gesturer talks about a ball. These gestures are subdivided into three sub-categories of Ekman and Friesen (1981) like spatial movements (i.e., depicting spatial relationships like distance, etc), kinetographs (i.e., depicting actions like movement of someone or something) and pictographs (i.e., depicting the form of objects). Similarly to iconic gestures, the metaphoric gestures illustrate the content of speech. However, they represent rather an abstract idea than a concrete

object or event. Metaphoric gestures include also *conduit gestures* which are iconic depictions of abstract concepts of meaning and language (McNeill, 1985). For instance, the speaker uses two open hands to make a spherical shape when saying: "That is a complete system!" to talk about the wholeness aspect of the system. The deictic gestures are pointing movements aiming at indicating an object that the gesturer is talking about. These gestures are also called pointing gestures. They are also subdivided into two sub-categories. The first sub-category refers to deictic gestures which indicate a concrete object or person. For instance, the speaker points with his index finger towards his interlocutor when saying: "It is you!". The second sub-category refers to gestures which indicate an abstract thing or object being mentioned in speech. For instance, the speaker uses a hand pointing first to the left when saying: "his idea is good" and then to the right when continuing: "but my idea is good also!". Beat gestures belong to a set of arbitrary gestures whose definition is: "A gesture is arbitrary when signal and meaning are linked neither by a relationship of similarity nor by any other relationship that allows one to infer the meaning from the signal even without knowing it". They are movements of arms or hands that go along with the rhythm of speech, but have no relation to the speech content. Such gestures are also called batons by Ekman and Friesen (1981). A typical beat gesture has only two movement directions such as in/out or up/down (McNeill, 1992).

Iconic, metaphoric and deictic gestures are considered as creative gestures (Poggi, 2002). They do not follow a standard form. We do not memorize each specific instance of them made in the mind, but we memorize a single and general rule to invent them on the fly. For instance, the rule proposed by Poggi (2002) for deictic gestures like "position your hand so as to describe an imaginary line going from your finger to the referent".

Other gesture types (i.e., non-communicative gestures) are out of scope of this thesis such as adaptor (i.e., self-touch like scratching one's earlobe (Kipp, 2005)), object manipulation (i.e., touching or reaching something (Butcher and Goldin-Meadow, 2000)) or emblems (i.e., conventionalized form and meaning that are mostly culture-dependent (Bitti and Poggi, 1991)).

2.3 Gesture Structure

In this section, we present a gesture structure proposed by Kendon (1972, 1980). Following Kendon, a gesture action includes one or more movement phases (i.e., preparation, pre-stroke-hold, stroke, post-stroke-hold and retraction). An overview of the gesture structure is illustrated in Figure 2.2.

- **Preparation** This optional phase starts a gesture: the limb moves away from a resting position to a position where the stroke phase will begin. At the same time of moving the limb, the hand configuration (e.g., shape, direction) is changed to be ready for stroke phase. There exists an unconscious temporal anticipation mechanism in this phase so that a gesturer starts a gesture at the right time to coordinate with other modalities (e.g., speech) (McNeill, 1992). This speech-gesture synchronization phenomenon will be presented in a later section.

- **Pre-stroke-hold** This phase is optional. When the position and the configuration of hands reach the end of the preparation phase, it may be held for a moment before the stroke begins. This phenomenon occurs because of several reasons. One reason is to allow attracting the attention of interlocutors and synchronizing gestures and speech (i.e., the gesture is delayed to wait for speech) (Kita et al., 1998).

- **Stroke** This phase is obligatory because it carries the most meaningful information of a gesture action that is related to the accompanied speech. The communicative shape and the trajectory of gesture movement is contained in this phase. Stroke phase is different from other phases in velocity and acceleration (Quek et al., 2002; Quek, 1995, 1994). For deictic gestures, which have two phases only, this phase is considered rather an independent hold without energy than a movement (Kita et al., 1998).

- **Post-stroke-hold** Similarly to pre-stroke-hold, this optional phase maintains the position and the configuration of hands at the end of the stroke phase. One reason is to attract the attention of interlocutors and synchronizing gestures and speech (Kita et al., 1998). Another reason is that there is no

retraction phase, due to a co-articulation between two consecutive gestures in a gesture unit (i.e., a gesture co-articulates to the next gesture).

- **Retraction** This phase ends a gesture: the limb returns from the end of a stroke to a resting position. Resting positions before the preparation phase and of this phase may be not the same. There exists, for instance, *partial retraction* and *full retraction* (Kita et al., 1998). The partial retraction means that the climb returns to a resting position in which the hand position and the hand shape are not yet at a fully muscle relaxing state. This phase is optional because it is not necessary for a gesture which has an articulated liaison to the next gesture.

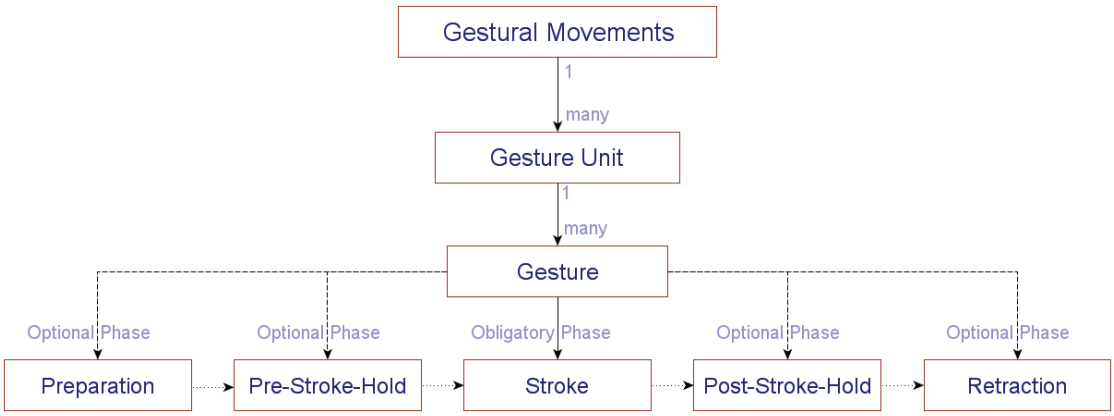


Figure 2.2: An overview of gesture structure

G-Unit A number of consecutive gestures are grouped into a gesture unit, called G-Unit (Kendon, 1980). Among these gestures, there are articulatory liaisons from one to another (i.e., a gesture avoids its retraction phase to continue to next gesture). Hence, a gesture unit starts to move the limb from a resting position and ends when it has reached a resting position again. In an utterance, there are several successive gesture units. These gesture units are separated from each other by resting positions.

2.4 Relations between gesture and speech

McNeill (1985, 1992) concludes that about 90% of all gestures occur during active speech. In this case, there are certainly some relations between gestures and speech. Following Kelly et al. (1999), gestures are related to speech firstly in semantic and temporal synchronization.

The semantic relation of speech and gestures Kendon (2004) claims that gesture is not inspired from speech, but gesture and speech come from the same original common intention. In other terms, gesture and speech are two aspects of the same process in which the gesture represents imagistic information and the speech represents propositional information of an utterance. In support of this claim, McNeill et al. (2000) notes that the two modalities have a close relation where they express the same communicative intent: one reinforces the other. For instance, gestures convey complementary, redundant information to the one indicated by speech.

We take one concrete case given by Bergmann et al. (2011) to illustrate the complementarity of a gesture to speech. In this example of gesture accompanying the utterance: "in the middle there's a clock", the gesturer describes the square shape of the clock by an iconic gesture using two hands. The information of the clock's shape is not mentioned in the speech. In this case the gesture complements the speech by describing the shape of the clock. In other words, the complementarity allows each modality to decrease the amount of information they carry and then, all information are merged to convey the same intention.

Regarding the redundancy between gestures and speech, the information being conveyed by speech is repeated again by a gesture. For example, a speaker says "the church is on the left" and positions the church with his left hand. This gesture is redundant with the speech because it does not send additional information in this case.

The temporal relation of speech and gestures According to Kendon (1980, 1972) and McNeill (1992), gestures and speech are produced by two separated processes but these processes are not independent: they are temporally related to

each other. In order to convey the same meaning at the same time, the execution of gestures has to be synchronized with the timing of speech and vice versa, the production of speech has to be coordinated with gesture timing. This phenomenon is considered as a mutual adaption between two modalities.

Firstly, the performance of gesture is adapted to speech timing. The stroke phase of the gesture slightly precedes or coincides with the stressed syllable of the speech. McNeill (1992); Ferré (2010) show that it exists an unconscious temporal anticipation in the preparation phase of a gesture so that a gesturer can start the gesture at right time to coordinate with speech.

Secondly, the performance of speech is adapted to a requirement of gesture structure. Following observations of McNeill (1992), the speech waits for the finish of the gesture preparatory phase. This phenomenon is concreted in the gesture speech generation model (i.e., growth points model) of McNeill et al. (2000) in which he indicates that there is an interaction between two processes of producing speech and of producing gesture so that they are able to influence one another.

Conclusion In conclusion, gesture and speech are two aspects of a same spoken utterance, the gesture represents the imagistic aspect and the speech represents the linguistic aspect. They convey the same communicative intent (i.e., semantic synchronization) at the same time (i.e., temporal synchronization). We follow these conclusions when implementing our system. While the semantic relation is taken into account in the process of gesture selection and in building gesture lexicon (i.e., gesture prototypes), the temporal relation between gesture and speech is considered in scheduling gesture to synchronize with speech.

2.5 Gesture expressivity

Gesture expressivity can be defined as a qualitative manner by which a gesture is executed, e.g., fast or slow, ample or narrow, smooth or hectic, etc.

In an experiment, Mancini et al. (2011) showed that if the expressivity of gestures is not changed from time to time, the emotional states cannot be transmitted effectively. They insisted that the expressivity has a link to emotion and it can convey an emotional content of behaviors. This observation was previously confirmed

in the experiments of Wallbott (1985, 1998) and (Wallbott et al., 1986) that show that the way of realizing gesture movements can be influenced by an emotional state. In their experiments, a video corpus of six actors (3 male, 3 female) who performed four emotions (joy, sadness, anger, surprise) was analyzed (Wallbott et al., 1986). The results indicate that emotions are not only displayed by facial expression, but also by body movement quality. Three factors of movement quality were defined: 1) movement activity; 2) expansiveness/spatial extension; and 3) movement dynamics/energy/power as illustrated in Table 2.1. The gesture expressivity refers to the manner of execution of gestures through such gesture quality factors. For instance, their experience showed that the sadness state is linked to low movement activity, spatial extended and powerful gestures whereas the angry state is expressed with high movement activity and high powerful gestures.

Movement quality factor	Sadness	Joy	Anger	Surprise
Activity (i.e., quantity of movement during a conversation)	low	medium	high	high
Expansiveness (i.e., amplitude of movement)	low	low	medium	high
Energy/power (i.e., dynamic properties of movement)	low	high	high	high
Pleasant (i.e., emotional properties of movement)	low	high	low	high
Fast (i.e., velocity properties of movement)	low	high	high	high

Table 2.1: Significant emotion effects for behavior cues (Wallbott et al., 1986)

Person's style is defined as a particular kind of expressing behavior that presents an individual's identity (Ruttkay et al., 2008). Gesture movement style indicates the way that person gestures. It is different from person to person and may depend on factors such as personality traits, gender and body types. From perceptive experiments, Gallaher (1992) characterized styles along four movement dimensions as illustrated in Table 2.2.

Dimension	Description
Expressiveness	Quantity and variation of movement
Animation	Factor "lethargic-animated" that is related to tempo-velocity
Coordination	Fluidity of consecutive movements (jerkiness vs. smoothness)
Expansiveness	Amount of space taken by the movement

Table 2.2: Individual differences in body movements: Dimensions of person's style (Gallaher, 1992)

Conclusion The expressivity in doing body movements (i.e., gestures) is linked to the personality or conveys the emotional states of a person. The levels of

movement expressivity are represented by modulating different dimensions of body movement such as the velocity, the expansiveness, etc.

2.6 Gesture Configuration

A gesture action is described through a sequence of hand movements which are specified by a set of properties. Following McNeill (1992), these properties can be divided into two parts: the first part describes the trajectory configuration (i.e., movement shape) and the second part describes the hand configuration at a specific time on the trajectory. The following subsections present their propositions as illustrated in Figure 2.3.

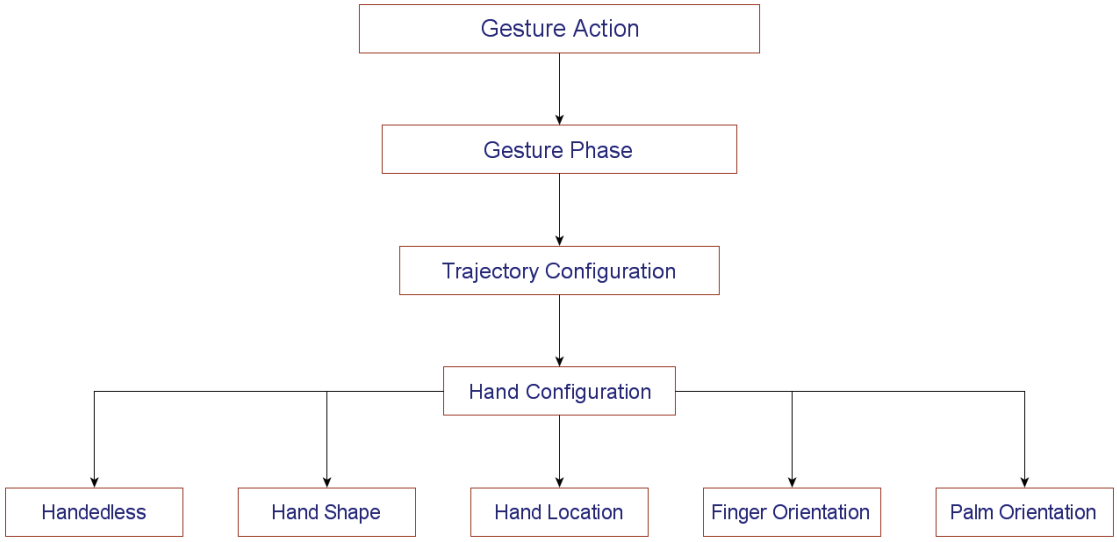


Figure 2.3: An overview of gesture configuration

2.6.1 Trajectory configuration

Spatial Representation This describes a gesture space in which hand gestures are formed and located. For instance, the gesture space proposed by McNeill (1992) is defined as a system of concentric squares around the gesturer. In this schema, the space is divided into many sectors in which each sector can be located by the hands. For instance, the sector directly in front of the chest is Center-Center as

shown in Figure 2.4. McNeill showed empirically that most gesture movements are realized in one gesture space only.

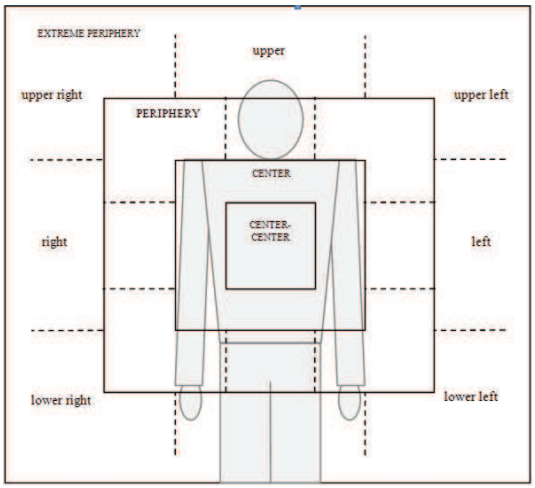


Figure 2.4: Concentric gestural space (McNeill, 1992)

Movement Primitives The movement of gesture includes the shape and direction of gesture trajectory. For a trajectory shape, for instance, several basic movement primitives are proposed like pointing, straight-line, curve, ellipse, wave, zigzag as illustrated in Figure 2.5.

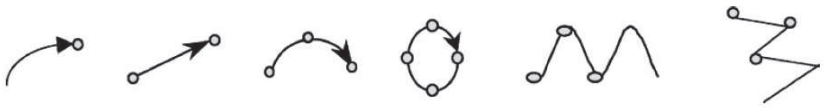


Figure 2.5: Basic movement primitives proposed by Gibet et al. (2001): from left to right: pointing, straight-line, curve, ellipse, wave and zigzag

A combination of two or more movement primitives forms more complex movements.

2.6.2 Hand configuration

The specification of hand configuration presented below is part of the Hamburg Notation System, called HamNoSys (Prillwitz, 1989).

- *Handedless* indicates which hand is used in gesture action: right, left or both hands.

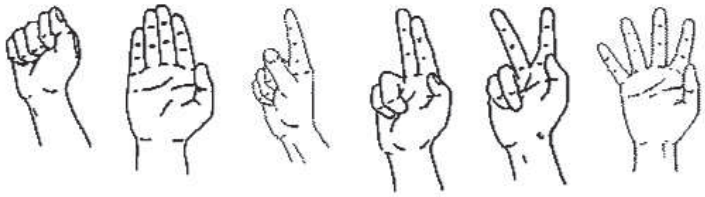


Figure 2.6: Some hand shapes from HamNoSys (Prillwitz, 1989)

- *Hand Shape* is formed by variations on the configuration of the fingers (i.e., thumb, index, middle, ring and little). Figure 2.6 illustrates some hand shapes which are defined in the Hamburg Notation System (Prillwitz, 1989).

- *Hand Location* The movement primitives describe the shape of gesture trajectories in which each trajectory is formed by a set of one or more hand configurations at key positions in a gesture space. Therefore, the values of hand location are attributed according to the gesture space.

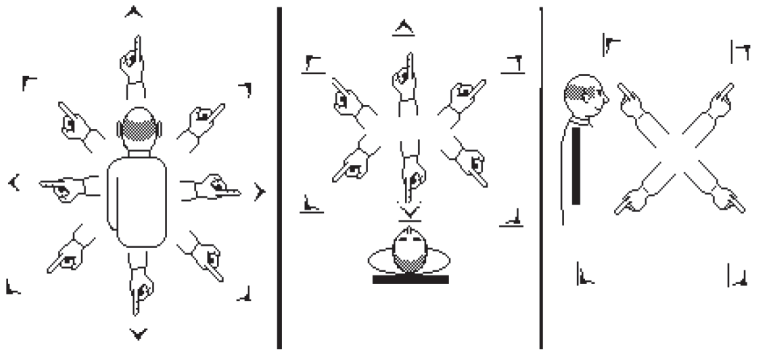


Figure 2.7: Directions defined for gestures from HamNoSys (Prillwitz, 1989)

- *Hand Direction* The orientation of a hand is described by the coordination of the orientation of extended fingers and of the palm. Figure 2.7 illustrates some possible directions which are defined in the Hamburg Notation System (Prillwitz, 1989).

Conclusion A human gesture can be described using a set of properties such as hand shape, hand location, palm direction, trajectory type. The definition of these properties is necessary to encode gestures.

2.7 Theoretical gesture generation models

Several theoretical gesture generation models have been proposed among which the model proposed by Levelt (1989). This model has been extended by different scholars. In this section, we will present two of such models: one model is developed by De Ruiter (1998) and another model is developed by Krauss et al. (2000). Many gesture systems are inspired from these theoretical models such as the gesture engine MAX in (Kopp et al., 2004a, 2008), Gesticulation Expression Model in (de Melo and Paiva, 2008), etc.

The similarity between these two models is that both of them are extensions of the speech production framework originally proposed by Levelt (1989). These extensions follow a conclusion that gesture and speech are two aspects of the same process: whereas speech conveys propositional information, gesture communicates imagistic information. Therefore, from the same input (i.e., an utterance) gesture and speech should have similar processes to generate corresponding signals. However there are still some differences between these two models that we will highlight.

The following subsections will firstly present the speech production framework of Levelt (1989) and then analyze two Levelt's framework based gesture models of De Ruiter (1998) and Krauss et al. (2000).

2.7.1 Levelt's model of speech production

The model of Levelt (1989, 1993, 1996) describes the production of speech. An overview of its architecture is illustrated in Figure 2.8.

Following this model, there are three main stages to produce speech in human: 1) firstly, conceptualizing message to communicate (i.e., idea or intention); 2) then formulating words corresponding to intentional message (i.e., lexicon and grammar) and afterwards planning how to utter these words (i.e., phonology); 3)

finally articulating planned words (i.e., overt speech).

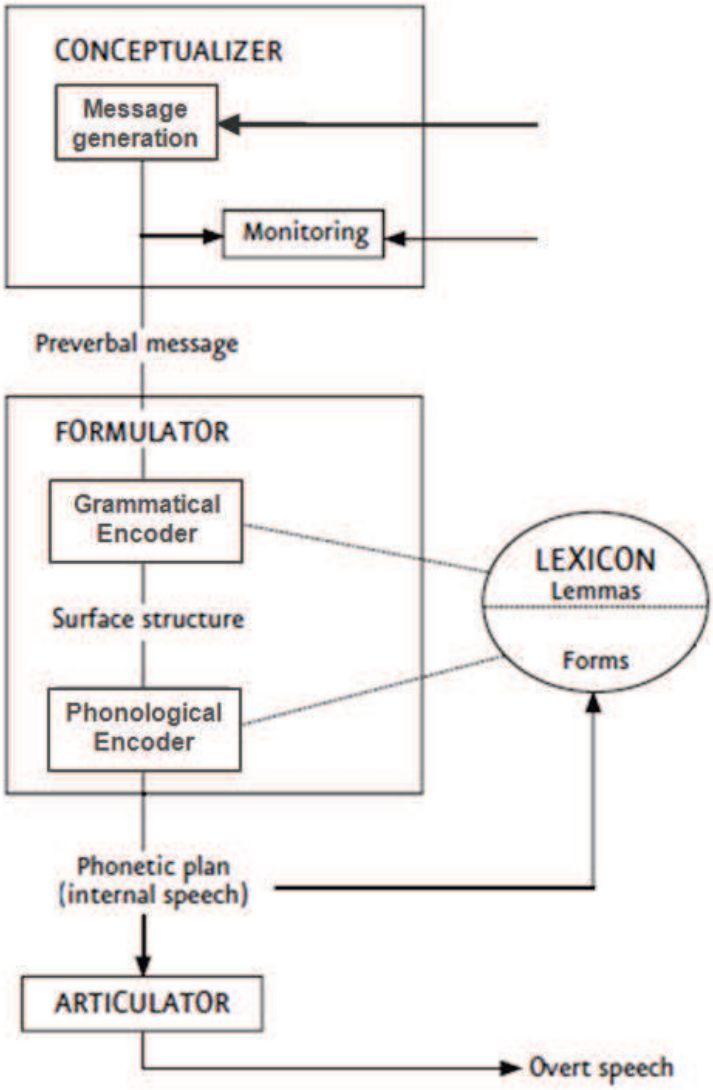


Figure 2.8: The speech production model of Levelt (1989): the boxes represent processing components and the circles represent knowledge stores

Conceptualizer First of all, a speaker must decide what he wants to express. In this stage, the main communicative intention to be conveyed is generated. A *conceptualizer* will encode planned preverbal message and send it to the next stage of the model.

Formulator This stage is divided into two sub-processes. The first sub-process (i.e., Grammatical Encoder) translates conceptual representation (i.e., preverbal message) into lexical representation. It retrieves lexical items or lemmas from a mental lexicon for the intended meaning. It means that it selects suitable words to convey a given intention and then plans how to organize grammatically these words in a correct way for utterance. The second process (i.e., Phonological Encoder) uses the results from the first process to plan sound units and intonation contour together to define how to utter the planned words, and then send them to the next stage of the model.

Articulator This stage receives the phonetic plan from the previous stage and executes it by muscles of the speech organs. The result of this stage is an overt speech.

In the Levelt's model, its processes focus on the speech production only. It does not consider non-verbal behaviors.

2.7.2 Sketch Gesture Model of De Ruiter

An extension of Levelt's speech production framework for a gesture generation model is proposed by De Ruiter (1998, 2000) (i.e., Sketch Model) as illustrated in Figure 2.9. Given a communicative intention, the first process, namely *Conceptualizer* accesses the working memory which contains hybrid knowledge representations to decide which information has to be conveyed in gesture and at which moment. This process is the same for both speech and gesture productions. It calculates propositional information for speech production and calculates imagistic (or spatial) information for gesture production. The second process, named *Gesture Planner* generates gestures from the corresponding sketch by retrieving an abstract gesture template from a gesture repertoire called *Gestuary*. The result of the Gesture Planner module (i.e. gesture sketch) is sent to the lower level as *Motor Control* modules where overt movements are planned and executed.

In this model, there are three different ways to generate gestures corresponding to three gesture types: 1) iconic gestures; 2) pantomimes gestures (i.e., symbolic gestures); 3) pointing and emblems gestures. Only pointing and emblems gestures

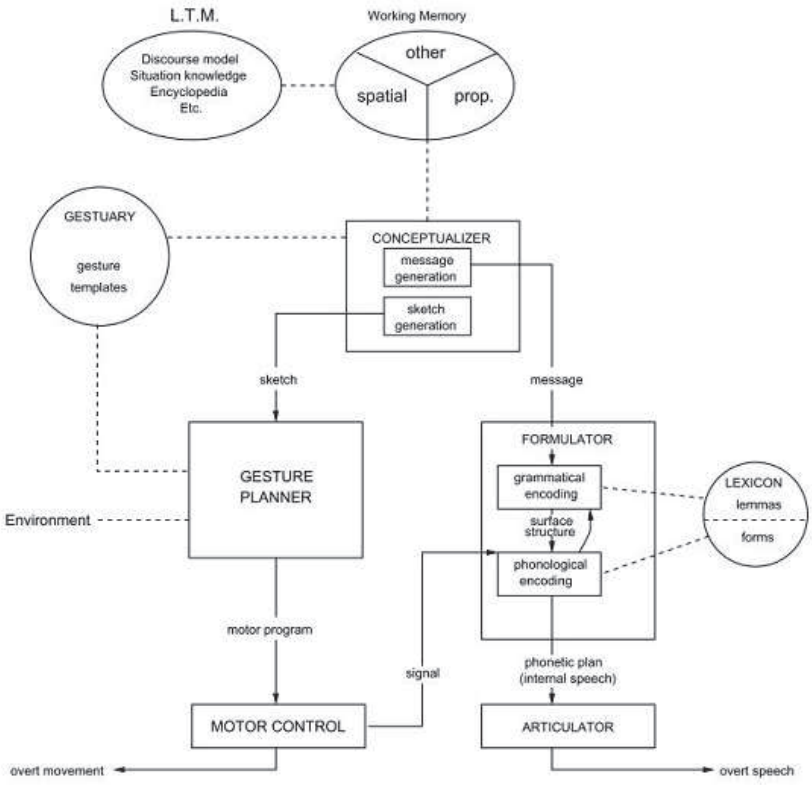


Figure 2.9: The Sketch gesture generation model of De Ruiter (1998): the boxes represent processes and the circles represent databases

are formed as abstract templates and stored in the Gestuary.

For the synchronization between gestures and speech, De Ruiter follows the findings of Morrel-Samuels and Krauss (1992) and Butterworth and Hadar (1989) that the start of gesture's stroke phase has to precede the stressed syllables of speech by a duration of less than 1 s and then the gesture can continue until after the finish of the speech. In addition, the timing of gestures and speech is adapted mutually. It means that a gesture could be filled by a gestural hold to wait for the accompanying speech to catch up. However, Levelt et al. (1991) showed that if the pre-stroke-hold phase lasts more than 300 ms, speech cannot adapt anymore because after 300 ms the speech is already active and cannot be changed. This is taken into account in this Sketch Model module. In the case of gesture stroke repetition, there is no gestural hold. Inversely, the timing of speech has to

adapt to the gesture timing too. Hence, speech and gestures are produced in two independent and parallel processes but these processes have to always exchange information to inform their states to each other.

The modules of Gesture Planner and Motor Control are two separate processes so that selected gestures can be realized by different gesture motor controllers.

2.7.3 Model of Krauss

Krauss et al. (2000); Krauss and Hadar (1999) extend the speech production model of Levelt (1989) for generating gestures. An overview of their gesture production model is showed in Figure 2.10. In general, this architecture also has three stages as in the model of DeRuiter. In the first stage, the process of *Spatial/Dynamic Feature Selector* calculates spatial representations that need to be described by gestures and output spatial/dynamic specifications. These output are abstract information involving trajectory direction and hand shape of a gesture. Then, these features are sent as data input to the Motor Planner module to be translated into a set of instructions for executing the lexical gesture (i.e., a motor program) (Krauss and Hadar, 1999). The process of motor system executes these instructions as gesture movements.

The synchronization mechanism between gestures and speech is ensured by adapting speech timing to the timing of gesture. It means that the gesture production process has to inform the speech motor once the gesture process terminates so that the system can adapt to the speech performance.

DeRuiter's model and Krauss's model differ in how gesture and speech are related to each other. DeRuiter follows the conclusion of Kendon (1980) and McNeill (1992) that gesture and speech represent two aspects of the same information process, both of them come from communicative intentions, while Krauss et al. (2000) follow the fact which is claimed by Rimé and Schiaratura (1991) that gestures come from memory representations and have functions to facilitate speech production. It means that the gestures in Krauss et al.'s model are not part of the communicative intentions. In their model, gestures are not generated from the Conceptualizer module (as described in the DeRuiter's model) but from a separate process called *Spatial/Dynamic Feature Selector*.

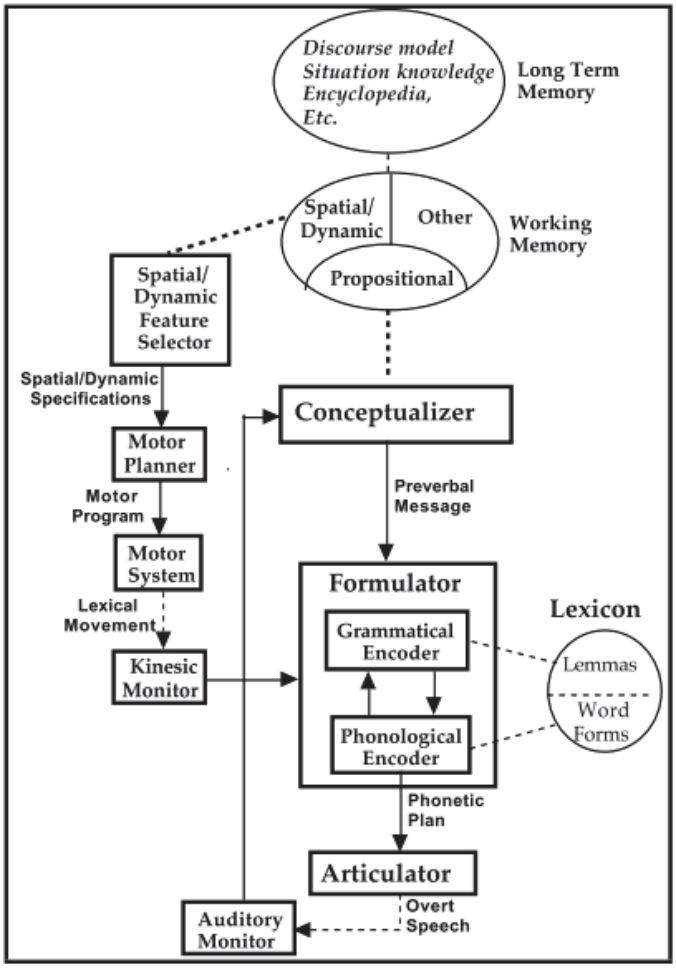


Figure 2.10: The theoretical gesture generation model of Krauss et al. (2000)

Conclusion

In conclusion, although the theoretical gesture production models of De Ruiter (1998) and of Krauss et al. (2000) have certain differences in processing the relationship between gestures and speech, they have a similar approach in creating gestures for a given intention: from the module of retrieving abstract gesture prototypes to the module of planning motor program. We follow these conclusions when designing our system: we elaborate also a repertoire of gesture templates (i.e., Gestuary) and use separated modules like Gesture Planner, Motor Planner in our system.

Chapter 3

State of The Art

The field of human-agent interaction attracts attention from researchers for its potential applications both in industry and in education. This field covers studies on virtual agents and on humanoid robots.

Virtual agents are defined as screen characters with a human body shape as illustrated in Figure 3.1. They have some intelligence to communicate autonomously with human users through verbal and nonverbal behaviors. A virtual agent is also called Intelligent Agent (IA) or Embodied Conversational Agent (ECA). In order to create a full realized agent, it must rely on diverse disciplines ranging from computer graphics, artificial intelligence, to sociology and psychology. In this Chapter, we present different approaches to generate gestures for virtual agents.

Figure 3.1: Some virtual agents display communicative gestures

A humanoid robot is a robot whose shape is designed to be resemble that of a human for certain objectives. One of these objectives is to have a physical

agent that communicates with us as if it is a human. To do that, a humanoid robot have to be equipped with the capability of communicating information via different channels such as speech, facial expressions, body movements, etc. In the limited study of this thesis, we focus on humanoid robot systems which can generate communicative gestures. Some of them are illustrated in Figure 3.2.

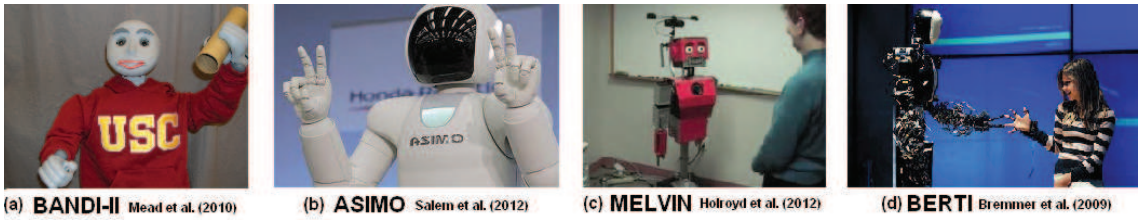


Figure 3.2: Some humanoid robots display communicative gestures

Recently, the advance of robotics technology brings us humanoid robots with certain behavior capacities as much as the virtual agents have (Holz et al., 2009). For instance the expressive anthropomorphic robot Kismet at MIT can communicate rich information through its facial expressions (Breazeal, 2003). The ASIMO robot produces gestures accompanying speech in human communication (Ng-thow hing and Okita, 2010). The Nao humanoid robot can convey several emotions such as anger, happiness, sadness through its dynamic body movements (Haring et al., 2011; Manohar et al., 2011). The similar capacities of those agents, virtual embodied agents (e.g., embodied conversational agents) and physical embodied agents (e.g. robots), allows researchers to use virtual agent frameworks to control robots (Salem et al., 2012).

This chapter presents some existing initiatives concerning gesture generation for a humanoid agent. It consists of three parts. The first part resumes different gesture generation systems for virtual agents; and the second part presents gesture engines for humanoid robots. The last part gives main differences between existing systems and our solution.

3.1 Gesture for virtual agents

In old cartoon films, motions including gestures for characters were created by a large number of consecutive static images (e.g., 25 images per second) drawn by hand or by the aid of computer graphics. Then with the development of motion capture technology, characters become more natural. Their motions are recorded from real human subjects as a set of small and complete animations. These animations are replayed in the game contexts.

However, using only prefabricated data or motion capture to feed for an animation engine has some limitations: it is difficult to produce animations which are not included in the original data. Hence, they are not appropriate for interactive virtual agents who need dynamic actions in realtime. In this section, we present agent systems which have certain features similar to our approach (i.e., creating dynamic gestures).

REA-BEAT

The first system that generates gestures for a virtual agent is proposed by Cassell et al. (1994) as illustrated in Figure 3.3. In their system, gestures are selected and computed from gesture templates. A gesture template contains a description of the most meaningful part of the gesture. These templates have no reference to a specific animation parameters of agents (e.g., wrist joint). The symbolic description of gesture templates allows their system to generate gesture animation dynamically in order to synchronize with phoneme timings of speech provided by a speech synthesizer. The gesture templates are elaborated manually and stored in a gesture repertoire called *gesture lexicon*. This method is still used in their later agent systems like REA (Cassell et al., 1999) and BEAT (Cassell et al., 2001). A similar approach is also used in our system. However, our model takes into account a set of expressivity parameters while creating gesture animations, so that we can produce variants of a gesture from the same abstract gesture template.

Figure 3.3: Gesture Speech Generation Model of Cassell et al. (1994)

SmartBody-NVBG

The system of Thiebaux et al. (2008a) uses the model-based approach to generate gestures for their SmartBody agent. The Smartbody system includes the NonVerbal Behavior Generator (NVBG) (Lee and Marsella, 2006) that uses some behavior generation rules to select and schedule gestures. The animation scripts returned by NVBG are encoded in Behavior Markup Language (BML) (Kopp et al., 2006) where multimodal behaviors are described symbolically (see the next Chapter for further details). A predefined animation corresponding to the description of gesture specified in the BML will be realized. For example, $<$ *gesture type* = "*you*" *hand* = "*righthand*" $>$ indicates that the agent uses right hand to do a "YOU" gesture. A predefined animation associated to the right-handed "YOU" gesture will be determined and played.

Data-model driven methods

While BEAT and SmartBody use model-based methods to reconstruct human gestures for their virtual agents, Stone et al. (2004) propose a data-driven method for synchronizing small units of pre-recorded gesture animation and speech. Their approach generates gestures synchronized with stressed syllable of speech automatically. Different combination schemes simulate agent's communicative styles. Another data-driven method is proposed by Neff et al. (2008). In this method, their model creates gesture animation based on gesturing styles extracted from

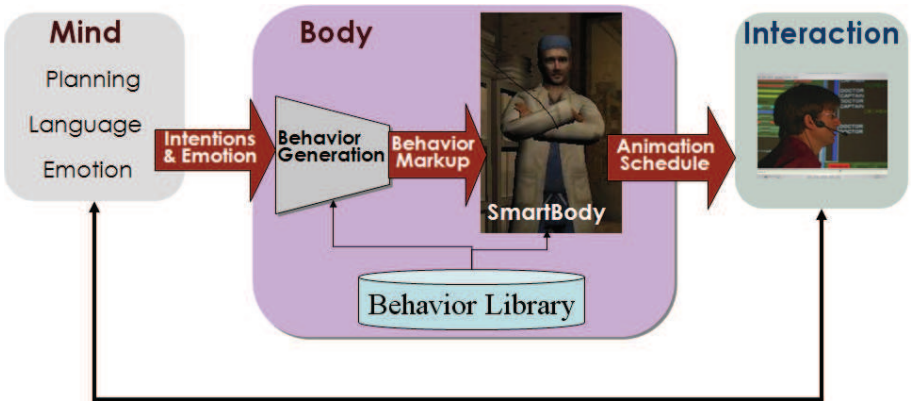


Figure 3.4: The SmartBody architecture (Thiebaux et al., 2008a)

gesture annotations of real human subjects.

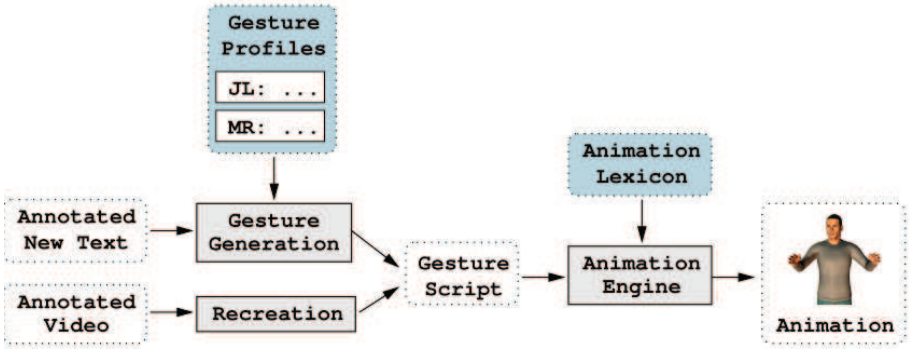


Figure 3.5: Personal profile based gesture generation system Neff et al. (2008)

GNetIc

A model that combines both model-driven and data-driven methods was developed by Bergmann and Kopp (2010). They applied machine learning techniques and rule-based decisions inherited from Bayesian decision networks, namely Gesture Net for Iconic Gestures GNetIc (Bergmann and Kopp, 2009) to select and form gestures which are linked to speech. The whole architecture is used for a computational Human-Computer Interaction simulation, focusing on the production of the speech-accompanying iconic gestures. This model allows one to create gestures on the fly. It is one of few models to have such a capacity. However, this

is a domain dependent gesture generation model. While our model can handle all types of gestures regardless specific domains, their automatic model is limited to iconic gestures. The model has to be re-trained with a new data corpus to be able to produce appropriate gestures for a new domain.

MAX-ACE

The GNetIc of Bergmann is integrated into the multimodal behavior realizer system of Kopp et al. (2004b) to generate automatically gestures for their virtual agent named MAX, a Multimodal Assembly Expert agent (Kopp et al., 2003). In this system, the results returned from the GNetIc module is an animation script encoded with MURML, a multimodal representation markup language (Kopp, 2005). Then, the Articulated Communicator Engine (ACE) module of their system interprets the MUMRL message and schedules gestures and speech so that they are synchronized with each other (Kopp et al., 2004a).

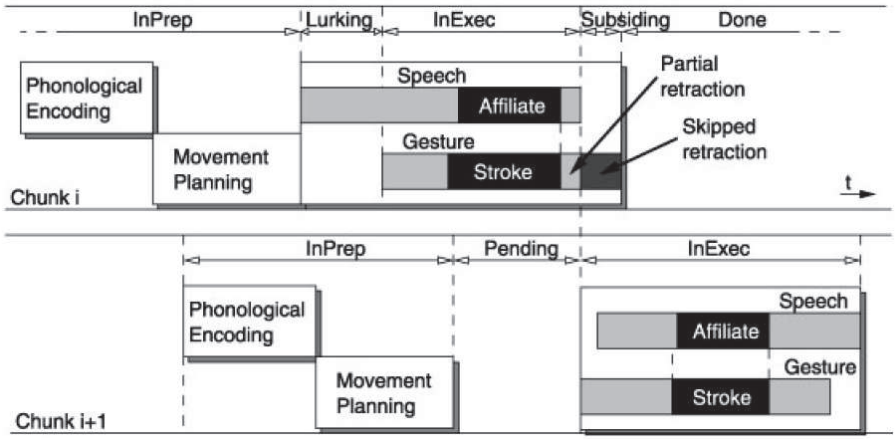


Figure 3.6: ACE: A gesture-speech scheduler engine (Kopp et al., 2004a)

A particular feature of the ACE engine is that the timing of gestures and of speech can be adapted mutually (Kopp et al., 2004a). The idea behind this engine is that an utterance can be divided into many chunks. A chunk of speech-gesture production is a pair of speech and a co-expressive gesture. In a chunk, these two modalities are temporally synchronized in such a way that the stroke phase of gesture and the affiliated words finish at the same time. Therefore, the duration

of gesture preparation phase is planed depending on their relative timing to the speech. Also, the start timing of speech is modulated with flexibility to adapt the required duration of the gesture (e.g., inserting a silent pause). This plan is ensured by an incremental scheduling algorithm as illustrated in Figure 3.6. In the process of ACE, the planned timing of modalities in a chunk cannot be re-scheduled when they are already queued to be realized. It means that there is no continuous adaption after a chunk is initialized.

Elckerlyc

Elckerlyc developed by Welbergen et al. (2010) is a continuous multimodal behavior generation system. It consists of several mechanisms that allow the interruption or the re-scheduling of ongoing behaviors with behaviors coming from a new demand in real-time while maintaining the synchronization between multi-modal behaviors (Reidsma et al., 2011). This system receives and processes a sequence of demands continuously allowing the agent to respond to the unpredictability of the environment or of the conversational partner. Elckerlyc is also able to combine different approaches to make agent motion more human-like when generating an animation. It uses both: a procedural animation and a physically-based simulation to calculate temporal and spatial information of the agent motion. While physical simulation controller provides physical realism of the motion, procedural animation allows for the precise realization of specific gestures.

AsaRealizer

The collaboration of Kopp and Welbergen et al. gives a new multi-modal behavior realizer system, named AsaRealizer (Van Welbergen et al., 2012). This system is built upon the Elckerlyc and ACE frameworks in order to benefit features of both their approaches such as the incremental scheduling algorithm of the ACE engine and the continuous interaction processes of the Elckerlyc system. Figure 3.7 illustrates an overview of this AsaRealizer system.

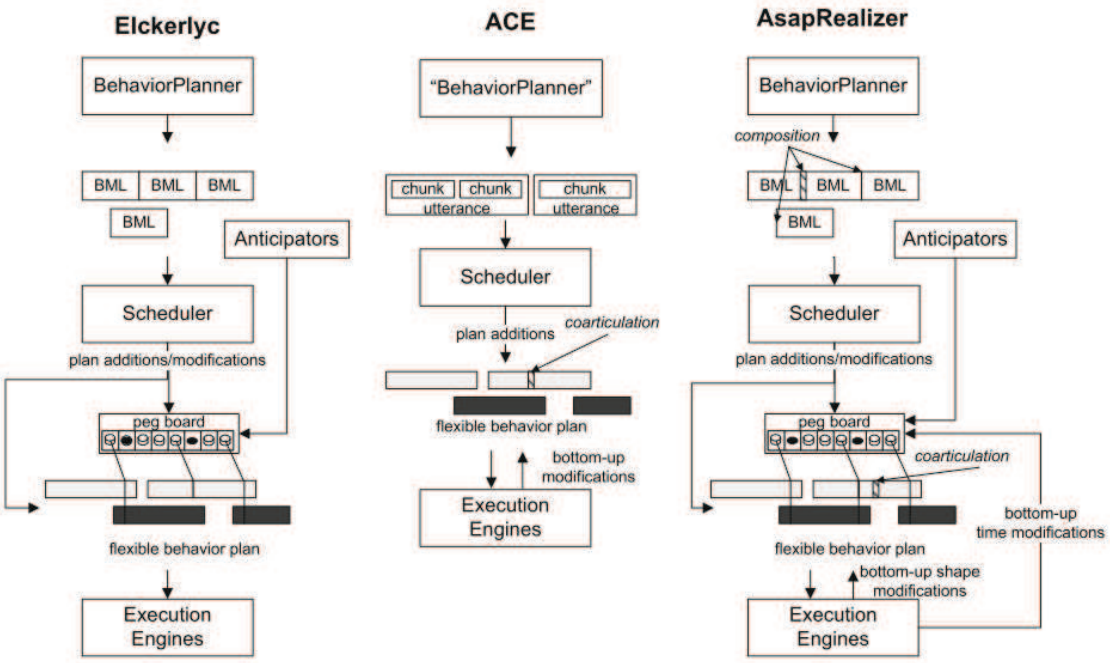


Figure 3.7: AsapRealizer: A combination of ACE and Elckerlyc (Van Welbergen et al., 2012)

EMBR

All the above presented systems have a common limit. They do not separate the higher-level control layer (e.g., the behavior scheduling process) from the animation parameters computing and playing processes (e.g., the animation engine). Such an architecture restrains the development of these systems when a new animation technology is applied. To solve this problem, Heloir and Kipp (2010a) proposed to add a new intermediate layer called *animation layer* into a behavior realizer system (see Figure 3.8) in order to increase animation control while keeping behavior description simple. Their real-time system offers a high degree of animation control through the EMBRScript language (Heloir and Kipp, 2009). This language permits us to control over skeletal animations, morph target animations, shader effects (e.g., blushing) and other autonomous behaviors. Any animation in an EMBRScript message is defined as a set of key poses. Each key pose describes the state of the character at a specific point in time. Thus, the animation layer gives an access to animation parameters related to the motion generation procedures.

It also gives an ECAs developer the possibility to control better the process of the animation generation without constraining him to enter the implementation details.

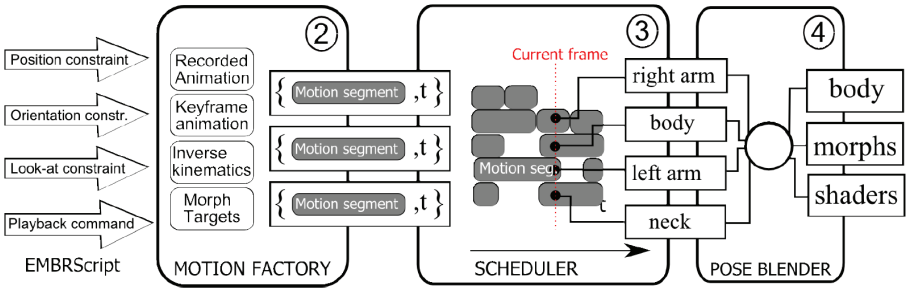


Figure 3.8: The architecture of EMBR system (Heloir and Kipp, 2009)

3.1.1 The common SAIBA framework

SAIBA (Situation, Agent, Intention, Behavior, Animation), an international standard multimodal behavior generation framework for a virtual agent (Kopp et al., 2006) as illustrated in Figure 3.9. This framework consists of three separated modules: Intent Planner, Behavior Planner and Behavior Realizer as illustrated in Figure 3.9.

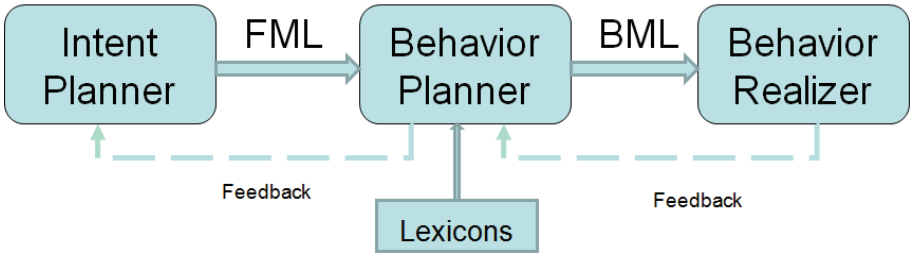


Figure 3.9: The SAIBA framework (Kopp et al., 2006)

The first module of Intent Planner defines communicative intents that an agent wants to convey in order to: i) express its emotional states, its beliefs; ii) describe its current goals; iii) respond to its interlocutor's feedback. The second module of Behavior Planner selects and plans multimodal behaviors corresponding to the given communicative intents. The last module of Behavior Realizer displays and

synchronizes the planned verbal and nonverbal behaviors of the agent. The results of the first module is the input of the second module through an interface described in Function Markup Language (FML) (Heylen et al., 2008) which encodes intentions and emotional states to be communicated by the agent. The output of the second module is encoded in Behavior Markup Language (BML) (Vilhjálmsson et al., 2007) and sent to the third module. Both FML and BML are XML-based script languages and independent from a specific agent. The SAIBA architecture includes also a feedback mechanism necessary to inform its modules about the current state of the generated animation. For instance, these information are used by the Intent Planner module to replan the agent's intentions when an interruption occurs.

3.1.2 Gesture Expressivity

Concerning the expressivity of nonverbal behaviors (e.g., gesture expressivity), it exists several expressivity models either acting as filter over an animation or modulating the gesture specification ahead of time. The EMOTE model implements the effort and shape components of the Laban Movement Analysis (Chi et al., 2000). These parameters affect the wrist location of a virtual humanoid agent. They act as a filter on the overall animation of the agent.

Other works are based on the motion capture to acquire the expressivity of behaviors during a physical action like a walk or a run in (Neff and Fiume, 2006).

On the other hand, a model of nonverbal behavior expressivity has been defined in such a way that it acts on the synthesis computation of a behavior (Hartmann et al., 2005b). This model is based on perceptual studies conducted by Wallbott (1998) and Gallaher (1992). Among a large set of variables that are considered in the perceptual studies, six parameters (Hartmann et al., 2006) were retained and implemented in the Greta ECA system. Their idea is to use this set of expressivity dimensions to modulate how gestures are executed. Variations in their parameters may come from factors such as current emotion, personality or context (Figure 3.10). One difference from the EMOTE system is that in the GRETA system gesture expressivity is taken into account at creating animations while in the EMOTE system the expressiveness is added on to existing animations. In this thesis, we

extended and implemented the set of expressivity parameters of Hartmann et al. (2006) to be able to control also the Nao humanoid robot.

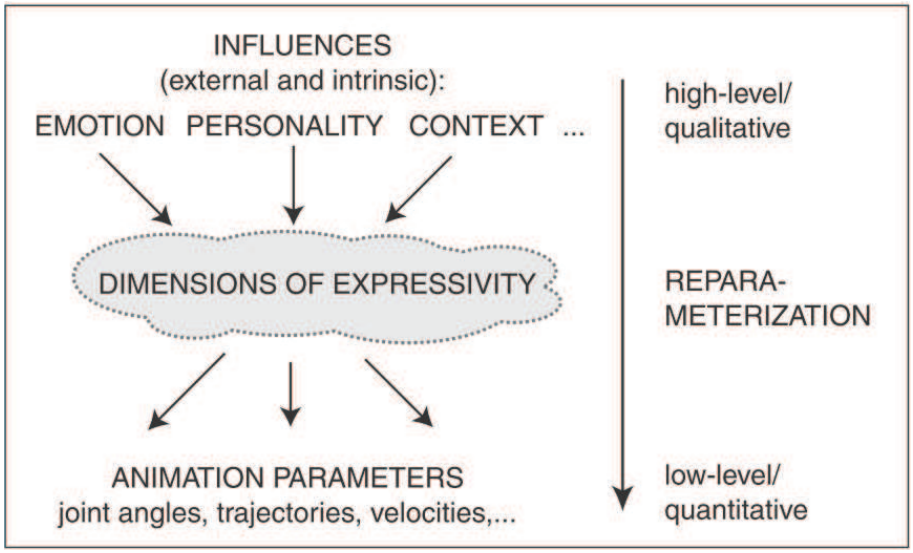


Figure 3.10: A mapping from high to low level agent functions of Hartmann et al. (2005b)

3.2 Gestures for humanoid robots

There are several ways to generate gestures for a humanoid robot. One way is to use predefined behaviors. For instance Monceaux et al. (2011) elaborate a set of gesture movement scripts using their behavior editor called *Choregraphe* (Pot et al., 2009) as illustrated in Figure 3.11. This method is limited as motion scripts are fixed and dedicated to a specific robot (e.g., Nao).

Another way to generate gestures for humanoid robots is to use basic gesture primitive movements. For example, Xing and Chen (2002) propose to compute their robot puppet's expressive gestures by combining a set of four movement primitives: 1) walking involving legs movement; 2) swing-arm for keeping the balance while walking; 3) move-arm to reach a point in space; and 4) collision-avoid to avoid colliding with wires. A repertoire of predefined gestures, called *Gesture Representation* is built by combining primitives sequentially or additionally. An overview of their architecture for primitives-based gesture production is shown in

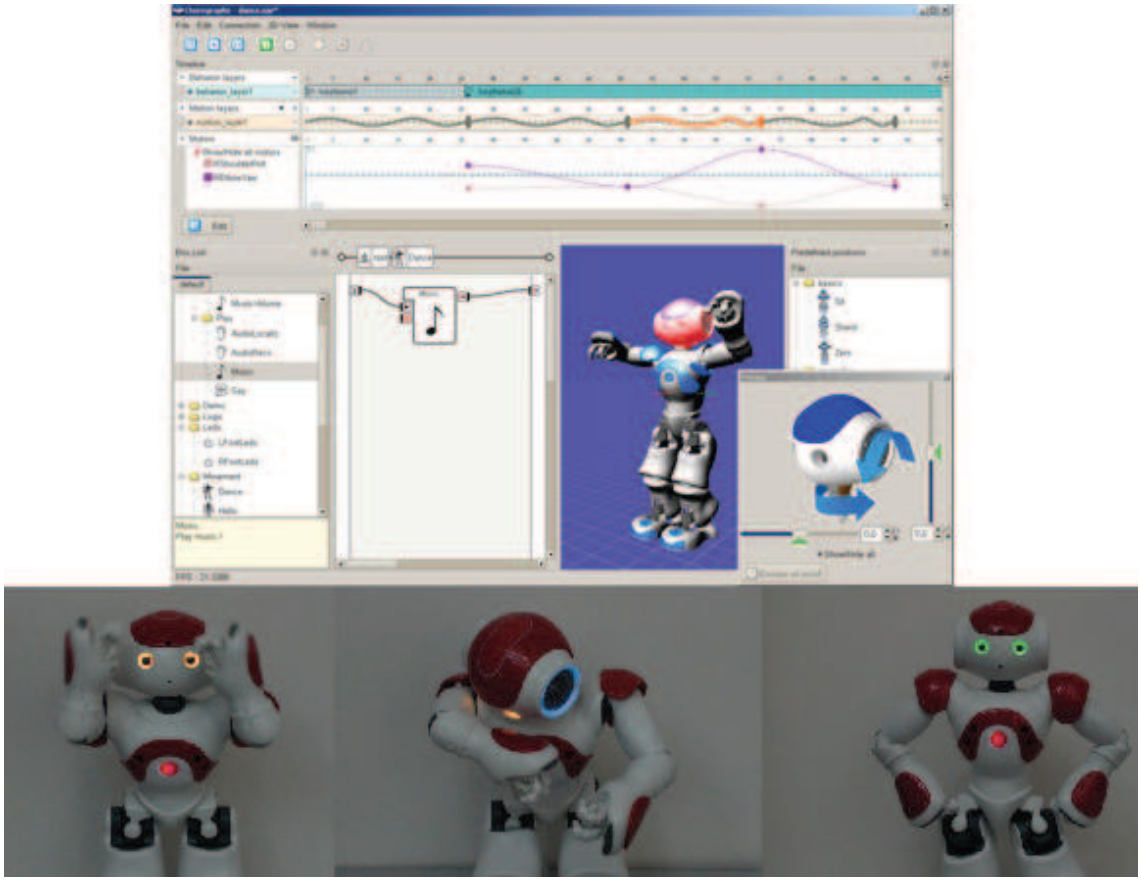


Figure 3.11: Choregraph behavior editor and some script-based Nao animations made by Monceaux et al. (2011)

Figure 3.12. In this architecture, given a stimuli received from the environment via a sensor (*Sensor Filter*) one gesture in *Gesture Representation* is selected to plan desired gesture trajectory. Based on the dynamics and kinematics of primitive templates, the *Gesture Planner* module allocates available body parts and calculates their motor action sequences. The output of the *Gesture Planner* module are parameter values which are sent to the *Primitives* module for an execution.

Behavior primitives are also used in the project of Mataric et al. (1998); Mataric (2000) to control their robot. Three primitives that they used: 1) move to position in a Cartesian space; 2) modulate the angles of joints to achieve an arm posture; and 3) avoid collisions. Similarly to the method of Xing and Chen (2002), gestures in the method of Mataric (2000) are generated by combining these basic primitives.

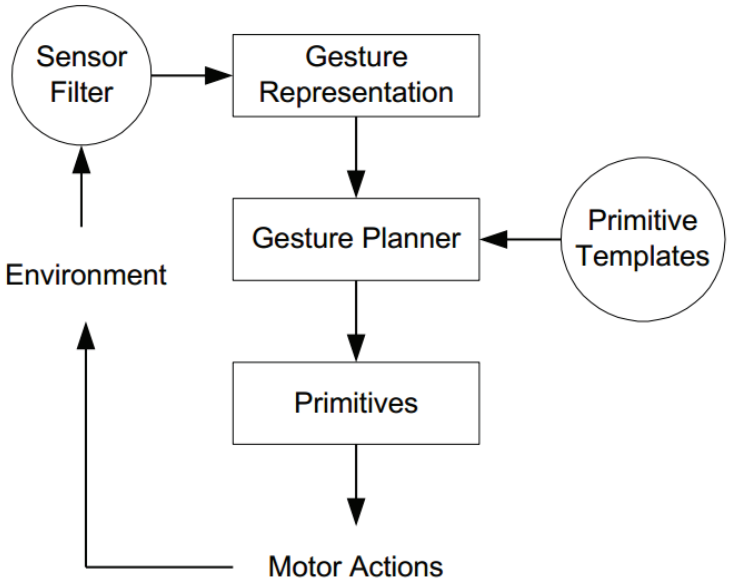


Figure 3.12: The architecture for primitives-based robot gesture production (Xing and Chen, 2002)

Gesture generation systems that are specific for a robot have also been developed. For instance, in the work of Bremner et al. (2009), the timing and the trajectory of a gesture action is calculated directly on specified motors of their robot.

Recent systems including our system apply a new approach that makes use of gesture generation solutions of virtual agents to endow humanoid robots with co-verbal gestures. For instance, Mead et al. (2010) integrates a subcomponent of the virtual agent system SmartBody (Thiebaux et al., 2008b), called *NonVerbal Behavior Generator* (i.e., NVBG), into their gesture robot system. Based on the NVBG rules their system selects gestures corresponding to the verbal content (Lee and Marsella, 2006) . However, the gesture realizer in their system is tied to a specific humanoid robot, the Bandit III robot (Figure 3.14a).

Holroyd and Rich (2012) implemente a system that interprets behavior scripts described in BML messages (Kopp et al., 2006) to control the Melvin robot (Figure 3.14c). This system follows using an event-driven architecture to solve the problem of unpredictability in the performance of their humanoid robot (Holroyd et al., 2011). Their system can equip the robot with simple gestures such as deictic

gestures, beat gestures accompanying speech to convey information in a Human-Robot Interaction context (e.g., in HRI games). The synchronization of gestures and speech is guaranteed by not only adapting gesture movements to speech timing but also adapting speech to gesture performance. Through a feedback mechanism, a pause can be added to speech to wait for a complex gesture to finish so that each pair of signals (i.e., gesture and speech) happens at the same time. However, their system is not designed to add an expressivity to the robot gesture animation.

An implementation and evaluation of gesture expressivity was done in the robot gesture generation system of Ng-thow hing and Okita (2010). This system selects gesture types (e.g., iconics, metaphorics, pointing, etc) corresponding to an input text through a semantic text analysis. Then, it schedules the gestures to be synchronized with speech using temporal information returned from a text-to-speech engine. The system calculates gesture trajectories on the fly from gesture templates while taking into account its emotion states. These parameters modulate the movement speed of gestures (e.g., happy or excited states accompany faster speeds and sad or tired states conduct slower speeds). Differently from our model, this system has not been designed to be a common framework for controlling both virtual and physical agents. Moreover, its gesture expressivity is concretely limited to one dimension parameter only (i.e., timing of movements).

There are also other initiatives that generate gestures for a humanoid robot such as the systems of Nozawa et al. (2004); Faber et al. (2009); Breazeal et al. (2005); Bennewitz et al. (2007) but they are limited to simple gestures or gestures for certain functions only. For instance, pointing gestures in presentation of Nozawa et al. (2004) or in Human-Robot Interaction experiments of Sugiyama et al. (2007); Shiomi et al. (2006).

The most similar approach to our model is the work of Salem et al. (2010b,a, 2012). They use the MAX system (Kopp et al., 2003) to control gestures of the ASIMO robot (Figure 3.14b). As the Max virtual agent and the ASIMO robot do not have the same behavior capabilities, they have to build a mapping from gesture descriptions created for the MAX agent to the ASIMO hand arm movements. The Articulation Communicator Engine (ACE) of the MAX system is modified to account for the smaller number of degrees of freedom (i.e., DOF) and kinematic dimensions of the robot.

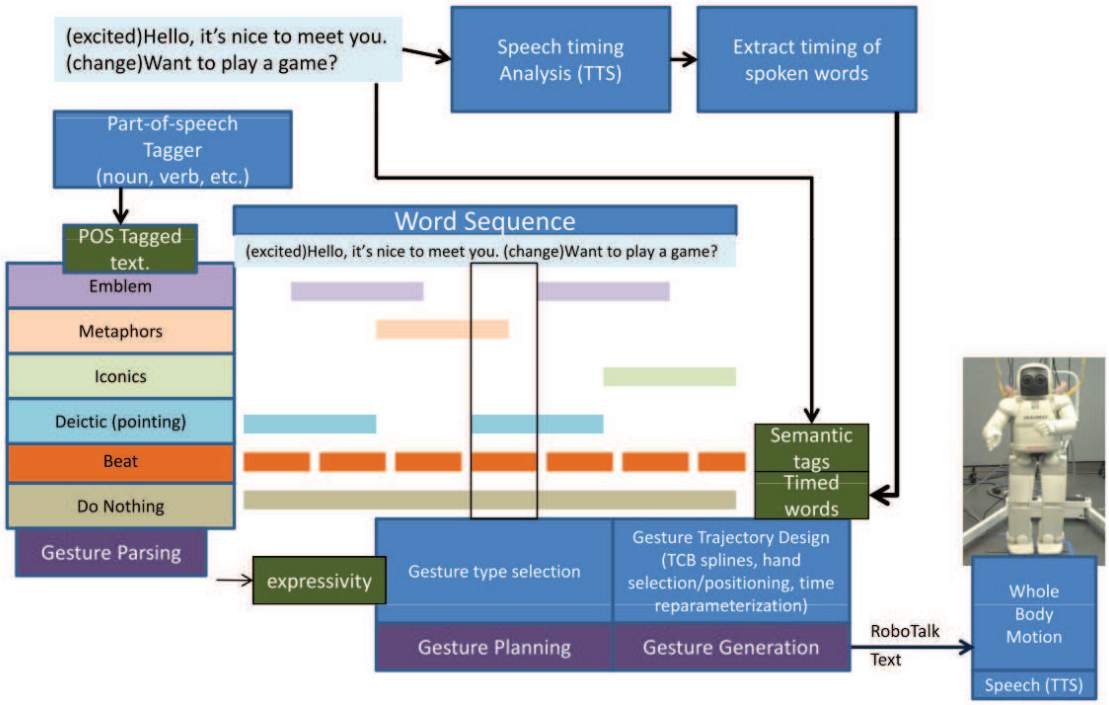


Figure 3.13: The robot gesture of Ng-thow hing and Okita (2010)

In addition to have all the features of the MAX system such as the automatic gesture generation and mutual coordination between gestures and speech, this system also has a feedback mechanism to receive information from the robot. The feedback module allows their system to modulate gesture trajectories in real-time which ensures that a gesture is finished in due time.

Regarding the work of Salem et al. (2010b), we share the same idea of using an existing virtual agent system to control a physical humanoid robot. Physical constraints have to be considered while creating robot gestures (e.g., limits of the space and the speed of robot movements). However, we have certain differences in resolving these problems. While Salem et al. fully use the MAX system to produce gesture parameters (i.e., joint angles or effector targets) which are still designed for the virtual agent, our existing GRETA system is redesigned and developed so that its external parameters can be customized to produce gesture lexicon for a specific agent embodiment (e.g., a virtual agent or a physical robot). For instance, the MAX system produces an iconic gesture involving complicated

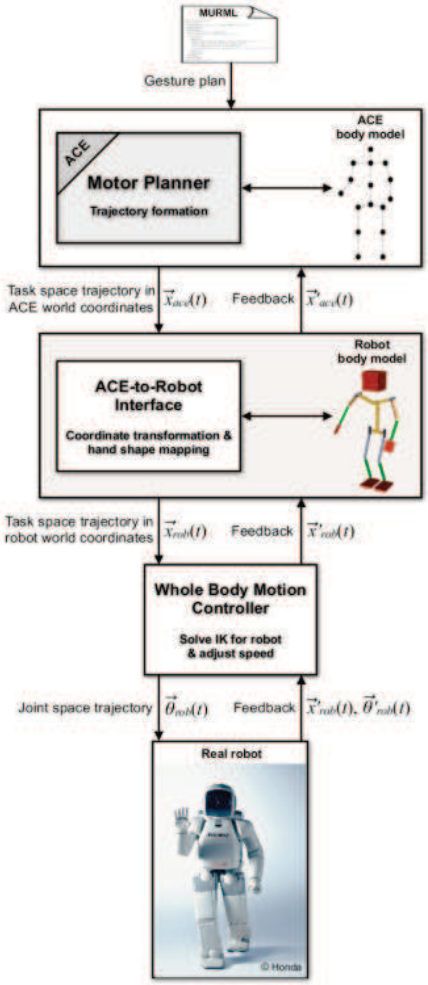


Figure 3.14: A similar approach to our model: Using the Max virtual agent to control ASIMO robot (Salem et al., 2010b)

hand shapes that are feasible for the MAX agent but have to be mapped to one of three basic hand shapes of the ASIMO robot. In our system, we deal with this problem ahead of time when elaborating the gesture lexicon for each agent type. In addition, the quality of our robot's gestures is increased with a set of expressivity parameters that are taken into account while the system generates gesture animations. This gesture expressivity has not yet been studied in Salem's robot system although it is mentioned in development of the Max agent (Bergmann and Kopp, 2010). The greatest advantage of the Salem's system is that it can

generate automatically gestures (Salem, 2012). Our system creates gestures on the fly from gesture templates regardless of a specific domain while their system uses gestures from a trained domain data corpus.

Summary

All of the above systems have a mechanism to synchronize gestures with speech. Gesture movements are adapted to speech timing in (Salem et al., 2010b; Ng-thow hing and Okita, 2010; Nozawa et al., 2004). This solution is also used in our system. Some systems have a feedback mechanism to receive and process feedback information from the robot in real-time, which is then used to improve the smoothness of gesture movements (Salem et al., 2010b), or to improve the synchronization of gestures with speech (Holroyd and Rich, 2012). They have also a common characteristic that robot gestures are driven by a script language such as MURML (Salem et al., 2010b), BML (Holroyd and Rich, 2012) and MPML-HR (Nozawa et al., 2004).

There are some differences between our system and the other systems. Our system follows the standard SAIBA framework of multi-modal behavior generation for humanoid agents. It is designed as a general framework so that its processes are independent from a specific agent. In our first experiment, our framework is used to control both the Greta virtual agent and the Nao robot without modifying its modules.

3.3 Conclusion

Our system shares common features with other virtual agent systems. For instance, all of four systems (SmartBody, Eclerkyc, EMBR and ours) are following the common SAIBA framework which has a modular architecture and whose animation is driven by the BML language. The symbolic description of gesture templates which only contains the *stroke* phase of gestures are used in BEAT, MAX and our system. These systems have also similar multimodal synchronization mechanisms which adapt nonverbal behaviors to speech timing. In particular, we share the same idea with Heloir and Kipp (2009) in adding a keyframe layer in the Behav-

ior Realizer module to facilitate the integration of an animation module in the global system. However, one important feature of our framework that is different from other systems is that our system can control different expressive agents from the same processes (i.e., using different external expressivity parameters). Table 3.1 shows some main differences between our framework and other virtual agent systems.

Authors	Expressive Behavior	Symbolic Lexicon	Virtual Robotic	SAIBA Compatible	Framework Name
Cassell et al. (2001)	No	Yes	No	Behavior Planer	BEAT
Thiebaux et al. (2008a)	No	No	No	Behavior Planner, Behavior Realizer	SmartBody
Kopp et al. (2003)	Yes	Yes	Yes	Behavior Realizer	MAX
Welbergen et al. (2010)	No	Non	No	Behavior Realizer	Elckerlyc
Heloir and Kipp (2010a)	Yes	Yes	No	Behavior Realizer	EMBR
Our system	Yes	Yes	Yes	Behavior Planner, Behavior Realizer	GRETA

Table 3.1: Comparison between different virtual agent systems

Authors	Gesture Expressivity	Speech-Gesture Synchronization	Gesture Template	Script Language	Robot Platform
Bremner et al. (2009)	No	Speech timing dependence	Fixed key points trajectory	No	BERTI
Shi et al. (2010)	No	Speech timing dependence	Fixed key points trajectory	MPML-HR	ROBOVIE
Holroyd and Rich (2012)	No	Mutual Synchronization	Limited to simple pointing gestures	BML	MELVIN
Salem et al. (2010a)	No	Mutual Synchronization	Automatic iconic gesture generation	MURML	ASIMO
Ng-thow hing and Okita (2010)	Modified trajectory and shape	Speech timing dependence	Basic trajectory shape of stroke	No	ASIMO
Our system	Modified trajectory and shape	Speech timing dependence	Basic trajectory shape of stroke	BML	Independent platform

Table 3.2: Comparison between different systems to generate robot gestures

With the high development of robotic technologies, it is possible to have humanoid robots with the same human-like appearance and behavior capability as the virtual agents. Although they have different embodiments (i.e., virtual and physical), both fields of virtual and robotic agents share the same knowledge background to model their expressive behaviors, both of them share the same issues of co-verbal gesture generation such as the gesture selection, the gesture description, the coordination between gestures and other behaviors, etc. That is why several research groups including ours had the idea of using the same platform

and same processes to control both of virtual and physical agents. So far, many robot systems have been developed but they are dedicated only to certain robots. Very few initiatives provide a general framework for different robots or provide a common framework that can be applied in both physical robot system and virtual agent system as we do. Table 3.2 summaries characteristics of the models we have surveyed.

Chapter 4

System Design

Recent research in the field of social robotics shows that, today it is possible to have robots with human-like appearance and expressive behavior capability just like virtual characters. A research question is proposed whether we can develop a common multimodal behavior generation system to control both virtual and physical agents. Our first approach relies on an existing virtual agent framework. For several years, some initiatives have been conducted to equip virtual agents with expressive behaviors. We use the embodied conversational agent platform GRETA (Pelachaud, 2005).

However, using a virtual agent system for a physical robot raises several issues to be addressed. Two agent platforms, virtual and physical, may have different degrees of freedom. Additionally, a robot is a physical entity with a body mass and physical joints which has certain a limit in movement speed. This is not the case of the virtual agent. The system have to take into account these constraints when nonverbal behaviors with expressivity are produced (e.g., their form and timing are planned).

This chapter presents our solution to design a common framework for generating multimodal behaviors, especially gesture signals for virtual and physical agents. The chapter includes three parts. The first part gives an analysis on gesture generation for a robot. The second part provides an overview of the existing GRETA system. The third part describes how the existing system is redesigned to control both agent platforms.

4.1 Gesture Generation Issues

In this section, we list all issues we have to deal with when implementing an expressive gesture model.

4.1.1 Difficulties

The synthesis of expressive gestures in particular for a real robot is constrained by its physical limitations. Communication is multimodal in essence. Speech, gestures and other nonverbal behaviors contribute to transmit information. The difficulty resides in specifying gestures for the robot which has limited movements (i.e., degrees of freedom). A symbolic gesture representation language is proposed that allows elaborating a repertoire of robot gestures. These gestures should be such that: 1) Their execution must carry a meaningful signification; 2) They are realizable by the robot (e.g., avoid any collision and singular positions).

The multimodal behaviors are coordinated and synchronized. They may be linked to each other by divers relations, such as a redundancy (e.g., showing the number "2" with two fingers while saying "two"), or a complementarity (e.g., opening the arms while saying "welcome"). The behavior generation model of the robot has to respect this rule of synchrony. It is necessary to develop a gesture scheduler within the model to ensure that: gestures while being executed are guaranteed to be tightly tied to speech uttered by the agent. A synchronization mechanism of behaviors for the robot is developed in consideration of its physical characteristics (e.g., limited speed).

In short, the gestures which are generated for a robot have to be conformed to constraints existing from its physical embodiment.

4.1.2 Gesture representation

Communicative expressive gestures convey information via their form (e.g., the wrist positions in space, the hand shape, the direction of palm and of extended fingers, the trajectory of wrists, etc) and their execution manner. Recently, an international initiative (i.e., SAIBA) has started working on a representation language for multimodal behavior (BML for Behavior Markup Language) to represent

and control the animation of a conversational agent. This representation is independent of the geometry and the animation settings of virtual agents. We want to use this language to control robot gestures. Despite the difference in degrees of freedom between a virtual agent and a robot, it is important that gestures of the robot and of the agent communicate similar information.

The manner of executing gestures is also linked to the gesturer's emotional states (Wallbott, 1985). To give a physical robot the gestural expressive capacity, it will be necessary to define how gesture expressivity is calculated. A predefined gesture prototype is combined with certain parameters of gesture quality such as timing, flexibility, etc. to generate a final expressive animation for the robot.

The difficulty resides in defining a gesture representation language to describe gestures in such a way that their surface form can be modulated in realtime.

4.1.3 Gesture expressivity

Six dimensions representing the expressivity of behaviors have been implemented in the Greta system based on the studies of emotional multimodal behavior of Wallbott (1998) and of Gallaher (1992). The expressivity dimensions were designed for communicative behaviors only (Hartmann et al., 2005a). Each dimension is implemented differently for each modality (Mancini and Pelachaud, 2008a). For the face, the expressivity dimensions refer principally to the intensity of muscular contraction and its temporality (i.e., how fast a muscle contracts). In the case of gestures, the expressivity works at the level of gesture phases (e.g., preparation, stroke, hold,...), and the level of co-articulation between many consecutive gestures. We consider the six dimensions which were defined in the work of Hartmann et al. (2005a). Three of them (spatial extent, temporal extent and power) modify the animation of a given gesture: The spatial extent changes the amplitude of movements (corresponding to physical replacement of facial muscle or of the hands); the temporal extent changes the duration of movements (corresponding to the speed of movement execution); and the power refers to dynamic property of movements (corresponding to the acceleration of movements). Another dimension, fluidity, specifies the continuity property of movements within a behavior signal or several consecutive signals of the same modality. The last two dimensions, global

activation and repetition, play on the quantity and the repetition of behaviors.

To ensure that gesture expressivity is realized by a physical robot, a mapping between gesture dimension parameters and robot effector parameters is developed. This mapping defines concretely how to replace a given gesture prototype by a corresponding expressive gesture.

The difficulty is whether the robot with physical constraints can convey correctly gesture expressivity via such a mapping function.

4.1.4 Gesture elaboration and production

The objective of this task is to identify gesture movements as gesture templates for an agent using the defined gesture representation language. An issue may rise when a robotic agent cannot do a gesture due to its physical embodiment (which a virtual agent can). This risk has to be foreseen. To do that the gesture model must take into account the constraints of the robot.

To calculate animation parameters, gestures are transformed into key poses. Each key pose contains joint values of the robot and the timing of its movement. The animation is script-based. It means that the animation is specified and described within a multimodal behavior representation languages. As the robot may have some physical constraints, the scripts are created in such a way that are feasible for the robot.

In this task, the difficulty resides in how to elaborate gestures which are significative and realizable for the robot.

4.1.5 Model of gestures-speech fusion

To develop a model of gesture-speech fusion, one must study three aspects: 1) the degree of redundancy or complementarity between gestures and speech; 2) the harmony of expressiveness between them; 3) and their temporal relationship.

While for the first issue, it is necessary to consider the content conveyed in gestures and in speech, the second issue requires a mapping between gesture expressivity and the emotional state of the robot (e.g., expressivity parameters for gestures for an anger state).

In the third issue, the study of synchronization between speech and gestures is an important activity because their temporal coordination determines the credibility of the expressive robot. The synchronization of gestures with speech is ensured by adapting gesture movements to speech timing. According to Kendon (1972) and McNeill (1992), the most meaningful part of a gesture (i.e., the stroke phase) mainly happens at the same time or slightly before the stressed syllables of speech. While a robot may potentially need longer time for the execution of wrist movements than a virtual agent, our synchronization engine should be able to predict gesture duration for each agent embodiment type so that their gestures are scheduled correctly.

From a technical point of view, with the high development of robotic technologies, it is possible that a humanoid robot (e.g., ASIMO, NAO) produces gestures and voice at the same time. A module should be developed so that it can simply control the robot movements and speech from a unified verbal and nonverbal multimodal behavior script (e.g., BML) indicating the surface description as well as the expressivity parameters of each signal.

Concerning this gesture-speech synchronization task, the difficulty resides in calculating the duration of gestures. Firstly, the gesture timing must be natural. Secondly, the velocity of gestures has to be limited within physical constraints of the robot (i.e., its speed limit).

4.2 The existing GRETA virtual agent system

Rather than developing a new gesture generation model, we rely on an existing multimodal behavior generation system, namely GRETA. This system has been developed for many years by a group of researchers and developers under the direction of Catherine Pelachaud (Hartmann et al., 2005a; Bevacqua et al., 2007; Mancini et al., 2008; Niewiadomski et al., 2009). The system equips an embodied conversational agent called Greta with the capacity of producing and responding appropriately with verbal and nonverbal behaviors like gaze, facial expressions, head movements and hand gestures to human users in realtime.

The existing GRETA system has modules compatible with the architecture of the SAIBA framework (Kopp et al., 2004b). The multimodal behaviors are con-

trolled by different abstraction levels from communicative intentions to animation parameters. The data flow is driven by four main modules: Listener and Speaker Intent Planner, Behavior Planner, Behavior Realizer, Animation Player (i.e, FAP-BAP Player). These independent modules receive and send input/output data via a whiteboard (e.g., the messaging system namely Psyclone allows modules to interact with each other through a local network). All of modules are temporally synchronized through a central clock. Following the SAIBA framework, the GRETA system also uses two representation languages FML and BML to communicate messages between different modules. However, the specification of these languages are extended to adapt to requirements of the GRETA system.

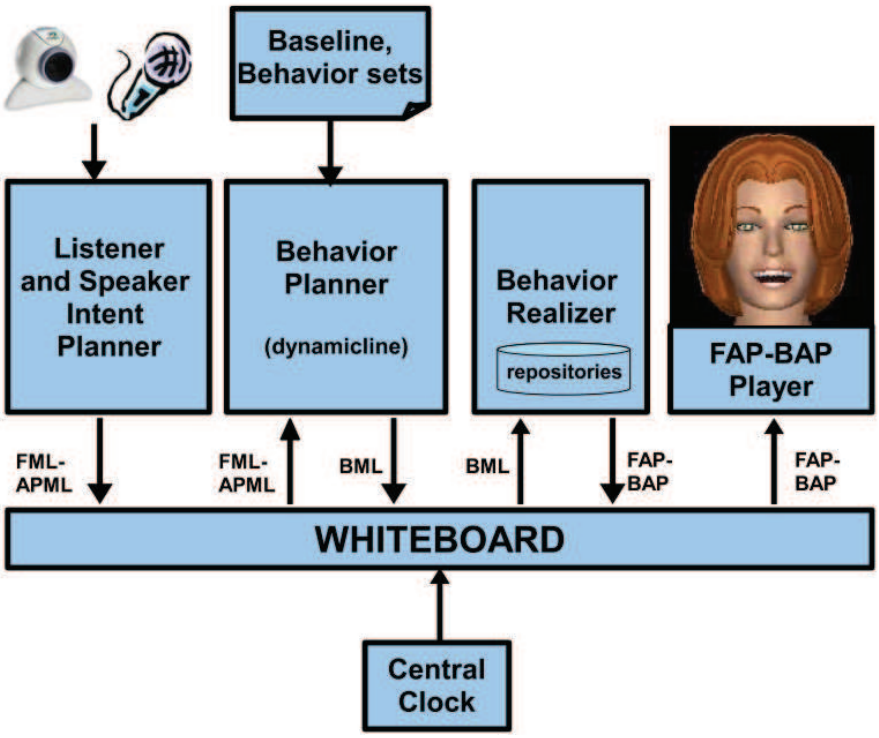


Figure 4.1: The existing architecture of the GRETA system (Bevacqua et al., 2010)

FML-APML

The FML language (Heylen et al., 2008) specifies communicative intents that an agent wants to communicate. This language has not yet been standardized.

Hence the GRETA system uses the FML-APML language proposed by Mancini and Pelachaud (2008b). The FML-APML specification is based on the Affective Presentation Markup Language (APML) (DeCarolis et al., 2004) and has similar syntax with FML.

Listing 4.1: An example of FML-APML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <!DOCTYPE fml-apml SYSTEM "fml-apml.dtd" []>
    <fml-apml>
        <bml>
                <speech id="s1" start="0.0">
                    <text>
                        <sync id='tm1'/> I <sync id='tm2'/> don't
                        <sync id='tm3'/> think <sync id='tm4'/> so!
                    </text>
                </speech>
        </bml>
        <fml>
            <performative id="id1" type="deny" start="s1:tm2" end="s1:tm3"
                importance="1"/>
        </fml>
    </fml-apml>
```

In general, a FML-APML message includes two description parts: one for speech and another one for communicative intentions as illustrated in Listing 4.1. The description of speech is borrowed from the BML syntax: it indicates the text to be uttered by the agent as well as time markers for synchronization purposes. The second part is based on the work of Poggi et al. (2004): it defines information on the world (e.g., location) and on the speaker's mind (e.g., emotional states, beliefs and goals). In this part, each tag corresponds to one of communicative intentions and different communicative intentions can overlap in time. Mancini (2008) defined a list of different tags used in the GRETA system such as *certainty* to specifies the degree of certainty of communicative intentions that an agent wants to express (e.g., certain, uncertain, double) or *emotion* to present agent's emotion states (e.g., anger, joy, fear), etc.

BML: Behavior Markup Language

BML is a XML-based representation language to describe communicative behaviors (i.e. speech, gestures, head movements, facial expressions, gazes, etc) with constraints to be rendered by an ECA. Its objective is to provide a general, player-independent description that is above a specific process implementation. In summary, its specification is to describe: 1) the occurrences of behavior; 2) the absolute or relative timing between behaviors; 3) the surface form of behaviors or references to predefined behavior templates; and 4) conditions, events or feedbacks, etc.

Listing 4.2: An example of BML

```
<bml xmlns="http://www.bml-initiative.org/bml/bml-1.0" id="bml1" characterId="
    Greta" composition="APPEND">
    <speech id="s1" start="0.0">
        <text>
            <sync id='tm1'/> I <sync id='tm2'/> don't
            <sync id='tm3'/> think <sync id='tm4'/> so!
        </text>
    </speech>
    <gesture id="g1" lexeme="DENY" start="s1:tm1" end="s1:tm4" >
        <description level="1" type="gretabml">
            <SPC.value>0.500</SPC.value>
            <TMP.value>0.000</TMP.value>
            <FLD.value>1.000</FLD.value>
            <PWR.value>0.000</PWR.value>
            <TEN.value>1.000</TEN.value>
            <OPN.value>0.400</OPN.value>
            <REP.value>1.000</REP.value>
        </description>
    </gesture>
    <head id="h1" lexeme="SHAKE" start="0.69" end="1.31" >
        <description level="1" type="gretabml">
            <SPC.value>0.000</SPC.value>
            <TMP.value>1.000</TMP.value>
            <FLD.value>0.400</FLD.value>
            <PWR.value>0.000</PWR.value>
            <TEN.value>1.000</TEN.value>
            <REP.value>1.000</REP.value>
        </description>
```

```
    </head>
</bml>
```

Listing 4.2 shows a BML message. In this example, there are three different signals. They are speech, gesture and head movements. The speech description is used to create audio data and calculate synchronized timings (i.e., time markers) returned from a speech synthesizer (e.g., the OpenMary TTS system developed by Schröder et al. (2006) or the Festival TTS system developed by Black et al. (1998) integrated within the GRETA system). The description of gesture and head signals does not specify a detailed surface form for them, but gives a reference to predefined prototypes namely "DENY" and "SHAKE" in their respective repositories. In this BML example the timing of gesture signal is relative to the speech through time markers, the absolute times are used for the head signal.

There are some extensions of behavior specification from the standard BML language in the GRETA system (i.e., the description within an extended level of BML messages). These extensions were defined by Mancini (2008) in order to represent expressivity parameters of nonverbal multimodal signals produced by the Greta agent. Thanks to these extensions, we can specify not only which signals the agent realize but also how these signals are realized.

Expressivity Parameters

We have a set of expressivity parameters for gesture signals which was defined and implemented for the Greta agent (Hartmann et al., 2005a, 2006). They are:

- *Spatial extent (SPC)* determines the amplitude of movements (e.g., contracting vs. expanding)
- *Fluidity (FLD)* refers to the smoothness and the continuity of movements (e.g., smooth vs. jerky)
- *Power(PWR)* defines the acceleration and dynamic properties of movements (e.g., weak vs. strong)
- *Temporal extent (TMP)* refers to the global duration of movements (e.g., quick vs. sustained actions)

- *Repetition (REP)* defines the tendency to rhythmic repeats of specific movements gesture)

Each expressivity parameter has value from -1 to 1 to represent the effect level of them on behavior performance in which 0 corresponds to a neutral state of behavior (i.e. normal movement). The combination of these parameters defines the manner to do a gesture signal.

Listener and Speaker Intent Planner

The Greta agent can play as a speaker or a listener alternatively. In the listener case, the Listener Intent Planner module is used. This module has been developed by Bevacqua (2009). It interprets signals received from its interlocutor like speech nuances or head movements to compute *backchannel* of the agent. The backchannel is defined as acoustic and nonverbal signals emitted from the listener to show his attention during conversation without interrupting the speaker's speech (Allwood et al., 1992). For instance, a head nod indicates that the listener agrees or understands what is said by the speaker, etc. The output of this module is a FML-APML message.

In case of being a speaker, the Speaker Intent Planner module is used. This module takes as input a text to be said by the agent provided by a dialog manager. The text is enriched with information on the manner the text ought to be communicated by verbal and nonverbal behaviors (i.e., with which communicative acts it should be realized). The output of this module is formatted within a FML-APML message.

Behavior Planner

The module Behavior Planner receives FML-APML messages from the Intent Planner module or from other sources (e.g., a compatible system developed by other researchers) via the whiteboard. After interpreting the messages, it selects, from an available nonverbal behavior lexicon, corresponding signals for conveying given communicative intents (Mancini and Pelachaud, 2007). A lexicon is made of pairs where one entry is an intention or emotional state and the other one is the set of behaviors that convey the given intention or emotional state. All possible agent's

communicative intentions, both while speaking and listening, are associated with multimodal signals that can be produced by the agent in order to convey them. Each one of these associations represents an *entry* of a lexicon, called *behavior set* (Mancini and Pelachaud, 2008a) as described in Listing 4.3.

Listing 4.3: An example of a behavior set for a virtual agent lexicon. The item within *core* tags mandatory to convey the "performative-greeting" intention.

```
<lexicon>
    <behaviorset name="performative-pointing">
      <signals>
        <signal id="1" name="smile" modality="face"/>
        <signal id="2" name="up" modality="head"/>
        <signal id="3" name="deictic=there" modality="gesture"/>
      <signals>
      <constraints>
        <core>
          <item id="3">
        <core>
    </behaviorset>
</lexicon>
```

One feature of this system is that it can generate distinctive behaviors for Embodiment Conversational Agents (Mancini, 2008). That is, a given communicative intent can be conveyed in different manners depending on the agent's baseline (as illustrated in Listing 4.4): 1) choice of lexicon; 2) set of preferred modalities (e.g., facial expression vs. gestures); 3) different values of expressivity parameters. The distinctiveness arises through the behavior selection process and their realization.

Listing 4.4: An example of Baseline

```
<modality name="gesture">
    <Parameter name="preference.value" value="0.1"/>
    <Parameter name="OAC.value" value="0.7"/>
    <Parameter name="SPC.value" value="0.4"/>
    <Parameter name="TMP.value" value="0.7"/>
    <Parameter name="FLD.value" value="-0.2"/>
    <Parameter name="PWR.value" value="0.3"/>
</modality>
```

```
<modality name="head">
    <Parameter name="preference.value" value="0.5"/>
    <Parameter name="OAC.value" value="0.7"/>
    <Parameter name="SPC.value" value="0.9"/>
    <Parameter name="TMP.value" value="0.3"/>
    <Parameter name="FLD.value" value="-0.3"/>
    <Parameter name="PWR.value" value="0.2"/>
</modality>
```

The output of this module is specified within a BML message containing a sequence of multimodal behaviors and their timing information.

Behavior Realizer

The main task of Behavior Realizer is to generate animation parameters from information received within a BML message. Firstly, it calculates the surface form of each signal from corresponding behavior repertoires. In these repertoires a list of prototypes of gestures, facial expressions, head movements, etc is stored and described symbolically. Then, these signals are scheduled from their relative or absolute timing information. Finally, animation parameters are computed while taking into account expressivity parameters for each signal.

In this module, an external text-to-speech synthesizer (e.g., OpenMary or Festival) is called to create sound data to be emitted by the agent. The TTS system also instantiates time markers and provides acoustic information necessary to synthesize lips movement.

Animation Player (i.e., FAP-BAP Player)

This module receives animations generated by the Behavior Realizer module and plays them on a computer graphic agent model (i.e., Greta). The system follows the MPEG-4 standard for creating and playing animation (i.e., Body Animation Parameters - BAP and Facial Animation Parameters - FAP). The animations are a sequence of FAP and BAP frames.

4.3 Using the GRETA framework for a robot

4.3.1 Defining problems

The GRETA system was originally designed for virtual agents only. At this thesis, we want to develop this system to become a general framework that works with different agents (i.e, virtual and robotic agents) by using common processes as much as possible. There are two main issues to be addressed in designing such a system: 1) the GRETA system did not take into account physical constraints of a robotic agent (e.g., a robot has less degrees of freedom and has some limits in its movement speed); 2) this existing system did not generate animation parameters for different embodiments (e.g., joint values of the Nao robot).

4.3.2 Proposed solution

Our proposed solution is to use the same representation languages (i.e., BML and FML) to control both virtual and robotic agents. This allows using the same processes for selecting and planning multimodal behaviors (e.g., gestures), but different algorithms for creating the animation.

The GRETA system calculates nonverbal behaviors which an agent has to realize for communicating an intention in a certain way. The system selects gestures taken from a repository of gestures, called Gestuary. In the gestuary, gestures are described symbolically with an extension of the BML representation language. Once selected, the gestures are planned to be expressive and to be synchronized with speech, then they are realized by the agent. After that, the gestures are instantiated as animation parameters and sent to an animation player (e.g., Greta player or Nao robot) in order to execute hand-arm movements.

In the following subsection, we present how the GRETA system is extended to be able to control different agents.

Solution for agents whose behavior capacities are different

Virtual and robotic agents may not have the same behavior capacities (e.g., the Nao robot can move its legs and torso but does not have facial expression and

has very limited hand-arm movements compared to the Greta agent). Therefore, the nonverbal behaviors to be displayed by the robotic agent may be different from those of the virtual agent. For instance, the Nao robot has only two hand configurations, open and closed; it cannot extend just one finger. Thus, to do a deictic gesture, it can make use of its whole right arm to point at a target rather than using an extended index finger as done by the virtual agent.

To control communicative behaviors of virtual and robotic agents, while taking into account their physical constraints, we consider two repertoires of gestures, one for the robot and another one for the agent. To ensure that both the robot and the virtual agent convey similar information, their gesture repertoires should have entries for the same list of communicative intentions. For instance, the index finger gesture in the Greta's repertoire and stretched arm gesture in the Nao's repertoire for the same pointing intention. In our proposed model, therefore, the Behavior Planner module remains the same for both agents and unchanged from the existing GRETA system. From a BML message outputted by the Behavior Planner module, we instantiate BML tags from either gesture repertoires. That is, given a set of intentions and emotional states to communicate, the GRETA system, through the Behavior Planner module, computes a corresponding sequence of nonverbal behaviors specified with BML syntaxes. A new Behavior Realizer module has been developed to create animation for both agents with different behavior capabilities. Figure 4.2 presents an overview of this solution.

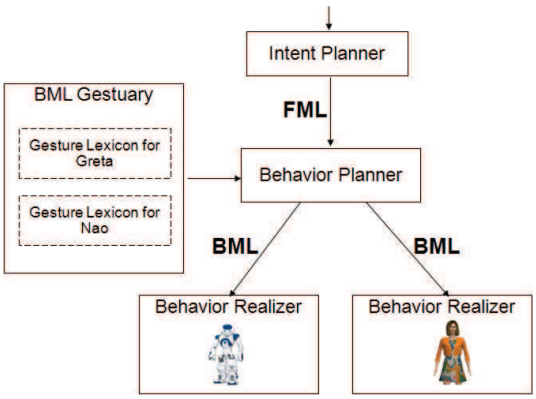


Figure 4.2: Solution: Agent-dependent lexicons for Nao robot and Greta agent

Solution for agents whose embodiments are different

We propose to develop agent-dependent animation generator modules: one for each agent. For instance, a FAP-BAP parameters generation module for Greta and a joint values generation module for Nao. In the existing GRETA system, the Animation Generator module was integrated within the Behavior Realizer module. Our proposed solution is to separate Animation Generator from the Behavior Realizer module in such a way that we can use Behavior Realizer as a common module for agents. To do that, an intermediate data layer is added. This layer is kept symbolically so that we can use the same processes to generate animation data (i.e., keyframes). This solution is illustrated in Figure 4.3.

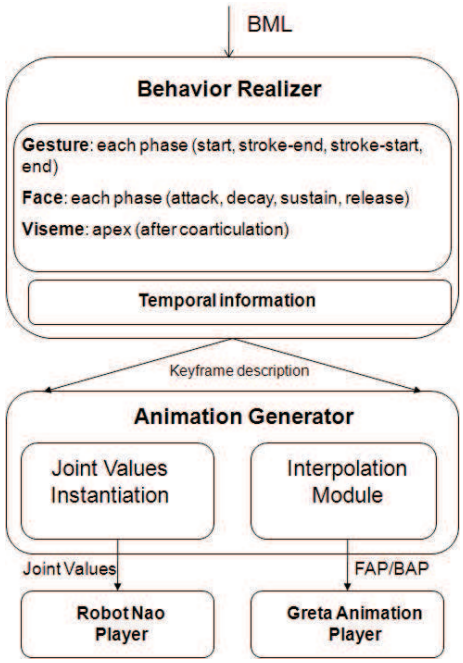


Figure 4.3: Solution: Agent-dependent animation generator for Nao and Greta

Each keyframe contains the symbolic description and the timing of each gesture phase (i.e., start, stroke-start, stroke-end, end, ...). Keyframes contain also other signals which are not mentioned in this thesis like facial phases (i.e., attack, decay, sustain, release) or viseme apex after co-articulation algorithm following Bevacqua and Pelachaud (2004). Keyframes are described using XML-based symbolical scripts. This symbolic representation allows us to use the same keyframes

generation algorithm for different agents from planned behaviors. It ensures that the processes of gesture selection and planning are independent from a specific agent embodiment or a specific animation parameters.

With this solution, our system architecture follows the theoretical gesture generation models of De Ruiter (2000) and of Krauss et al. (2000) mentioned in Chapter 2 (i.e., Background). Our Behavior Realizer calculates and schedules a gesture trajectory from gesture templates (i.e., gesture lexicon) corresponding to Gesture Planner of De Ruiter (2000) and Spatial/Dynamic Feature Selector of Krauss et al. (2000). The Animation Generator and Animation Player modules plan and execute overt movements respectively in our model corresponding to Motor Control of De Ruiter (2000)'s model and Motor Planner and Motor System of Krauss et al. (2000)'s model. This compatibility is illustrated in Figure 4.4. One difference between these two models and our model is that our architecture has been designed to be extended for the generation of multimodal nonverbal behaviors including facial expression, torso, head, etc. instead of only gesture and speech signals as in their model. Moreover, we introduce a more fine-grained nonverbal behavior processing which focuses on the synchronization and the expressivity of produced nonverbal behaviors.

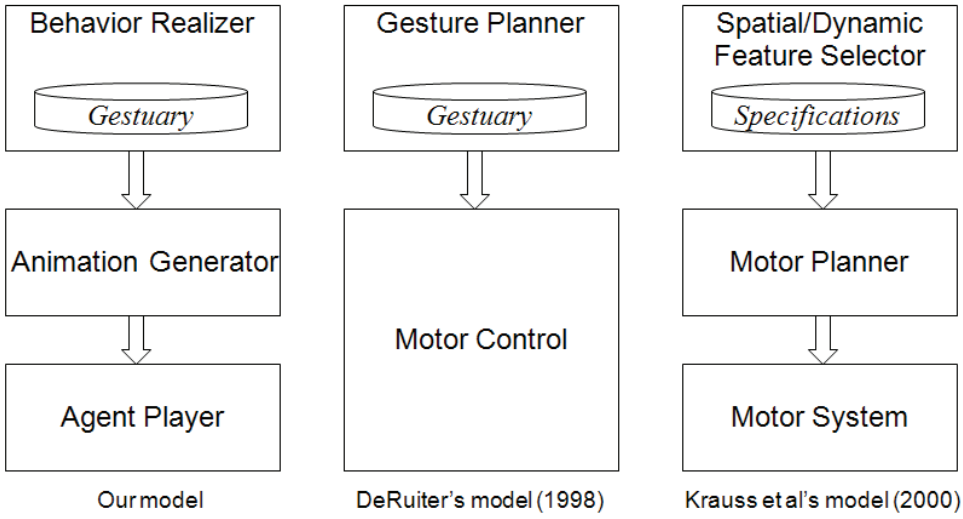


Figure 4.4: A comparison between our model and two other theoretical gesture generation models

4.3.3 A global view of the system architecture

In summary, following the proposed solutions, we present a new architecture of the GRETA system as described in Figure 4.5. In this architecture, three main modules of the system, Intent Planner, Behavior Planner and Behavior Realizer are used as common modules for agents. Only the module of Animation Generator is specific to an agent.

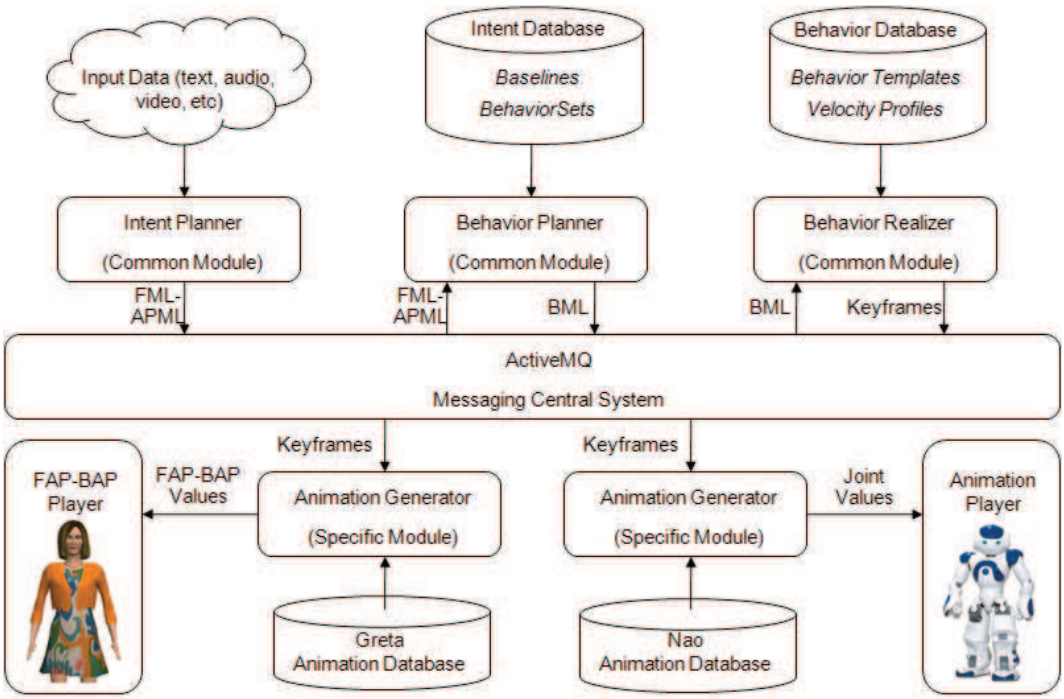


Figure 4.5: The overview of the new GRETA system

Among four main modules in the new architecture of GRETA, the first two modules (i.e., Intent Planner and Behavior Planner) were developed by Mancini (2008) and Bevacqua (2009) and remains unchanged from the existing system. The computation of communicative intents and the selection of nonverbal behavior signals (e.g., gesture signals) are beyond the scope of my thesis.

4.4 Toward a gesture expressivity model

Based on the given issues to be addressed for a physical robot and the proposed extensions of the GRETA framework for a general expressive gesture model, we need to address several issues at the theoretical and technical levels.

4.4.1 Research issues

The research work is related to studies of human gestures accompanying speech which are applied for a humanoid agent. We have to answer three main questions concerning gesture generation: 1) How to encode human gestures and reproduce these gestures which are realizable for agents; 2) How to schedule them to be synchronized with speech; 3) How to render them expressively. Our final objective is to have a computational model of expressive gestures. Answers of these research questions are addressed in three stages of our model. In the first stage, gestures are encoded in repertoires of gestures using a retrievable representation. In the second stage, gestures are calculated and then displayed by an agent. These gestures are evaluated by human users through a set of perceptive tests in the third stage.

To address these issues, we base on gesture studies presented in Chapter 2. For the first question, the structure of gestures is studied. A set of properties of a gesture is extracted in order to define a gesture representation language. The value of gesture properties should be symbolical so that the same syntaxes of the proposed representation language can be used to describe gestures for both virtual and physical agents. Additionally, symbolical values allow developing the same algorithms to process their gestures. Consequently, following the designed system architecture in the previous section, to overcome the issues of gesture realizability for a robotic agent, we can use this gesture representation language to elaborate two repertoires of gestures: one for the virtual agent and another one for the physical agent.

In the second question, we have to deal with a gesture trajectory that includes not only an unique gesture but also a sequence of consecutive gestures. It means that it exists a co-articulation between gestures in an utterance. Scheduling a gesture trajectory has to output information for two questions: 1) When does the

gesture trajectory start? and 2) How long does it take? The system has to rely on the relationship between gestures and speech to produce these information. Moreover, in order to have natural gesture movements, the human gesture timing has to be studied. The simulation of human gesture timing for a robot may be constrained by the maximal velocity of its joints. Then gesture velocity profile for each agent will be built to overcome this issue. In all cases, the temporal coordination between gestures and speech must be respected: The stroke phase of a gesture happens at the same time with stressed syllables of the speech.

For the third question, in order to render gesture animation expressively, the set of expressivity parameters defined by Hartmann et al. (2005a) are used. These parameters have been implemented for the virtual agent. Now we apply them for robot gestures. From a symbolical gesture prototype, the system generates different variants of the gesture by modulating its expressivity dimensions in realtime. The robot does not have a gesture movement space which is as free as the virtual agent's. A gesture movement space including key positions is defined for the robot. Hence, the spatial change of gestures of the robot is limited in its gesture space. For the similar issue, the temporal change of gestures is limited in robot gesture timing profile.

To validate the developed expressive gesture model, we conduct a perceptive experiment. In this experiment, we want to evaluate how robot gestures are perceived by human users at the level of the expressivity, the naturalness of gestures and the synchronization of gestures with speech while the robot is telling a story. Through this experiment, we can evaluate the quality of gesture expressivity for a combination of different dimensions (e.g., spatial extent, temporal extent, etc) as well as the quality of the temporal coordination between gestures and speech. The results of this experiment will answer the research question: "Whether a physical robot can display gestures with expressivity?".

4.4.2 Technical issues

The technical work lies in refining the Behavior Realizer and Animation Generator modules to realize expressive communicative gestures for different agents.

Behavior Realizer

The processes in this module are common to different agents. This module takes as input BML messages and external agent-dependent parameters like gesture repository and gesture velocity specification for reproducing and scheduling gestures. The results of this module is a set of keyframes.

In detail, this module separates the gesture database from its processes. While the database contains parameters specific to an agent (e.g., covering also constraints of each agent), the processes are the same for different agents.

The processes in this module take place in three stages. The first stage concerns a BML parser which validates received BML messages (e.g., avoiding infinite loop in case that signals are temporally depended on each other) and initializes signals to be computed (e.g., fill up the surface form of indicated signals from their repertoire). The second stage focuses on the creation and the scheduling of gesture trajectory while taking into account expressivity parameters. The third stage is responsible to generate a list of keyframes as mentioned in Section 4.3.2.

Technically, this module is developed in the programming language Java which is compatible with other modules in the GRETA framework.

Animation Realizer

The second one is an embodiment specific module whose objective is to generate animation parameters corresponding to an agent to be sent to the agent player. While my thesis work deals with an instantiation module of joint values for the NAO robot, another similar module to handle body animation parameter (BAPs) and facial animation parameters (FAPs) of the MPEG-4 standard has been developed by Huang and Pelachaud (2012) and Niewiadomski et al. (2012) for the Greta virtual agent.

In the Animation Realizer module, there are some algorithms dedicated to the Nao robot to overcome its particular limitations. Firstly, it exists singular positions in its gesture movement space that the robot hands can not reach. Secondly, it is impossible to estimate the duration necessary (i.e., a minimum duration) to do a robot hand movement before realizing this movement. To deal with these issues, we have to define a gesture movement space containing predefined reachable positions

following the gesture space of McNeill (1992). Moreover, the duration necessary to do a hand movement between any two positions in the defined space has to be calculated ahead of time. Gestures whose available time is not enough to be performed have to be canceled and leave time to next gestures.

Additionally, there are three requirements to be respected in developing this module. Firstly, the module has to synchronize temporally with the central clock of the global GRETA system. This ensures that a behavior is executed at the time as it was planned. Secondly, different behavior modalities have to be dealt in parallel processes so that they can be started at the same time if necessary. Thirdly, the processes have to be continuous so that a new coming keyframe can be integrated within previous keyframes.

Technically, this module is developed in the programming language C++ based on available APIs provided in the NAO Software Development Kit (Gouaillier et al., 2009).

Tests

Before implementing a perceptive experiment, the technical algorithms developed in these modules have to be verified and validated. A set of test cases is conducted to ensure: a) the algorithms work as expected; b) the system controls physical constraints of the robot.

Chapter 5

Implementation

Our expressive gesture model is developed within the GRETA nonverbal multimodal behavior generation framework as designed in Chapter 4. Many researchers and developers in our lab have contributed to the construction of the new framework (see Figure 5.1). However, in this chapter, we focus on modules for which we have contributed: a gesture database, a gesture engine integrated within the behavior realization module and an animation generation module for the Nao robot. The database is designed as a set of external parameters separated from their processes in order to be used by different agents.

5.1 Gesture Database

A gesture database is a structured collection of gesture data. The data are organized to model relevant aspects of human gestures (e.g., arm movement, hand shape, etc) in a way that supports processes requiring this information (e.g., reproducing a gesture to be expressive and to be synchronized with speech).

In our system, the gesture database includes two components: Gesture Templates Repository and Gesture Velocity Specification. These components have the same objective: providing the system with necessary data to produce gestures for various humanoid agents (i.e., virtual and robotic agents). They are used at different gesture computation stages in the Behavior Realizer module: 1) Gesture Template Repertoire contains abstract gesture templates providing sym-

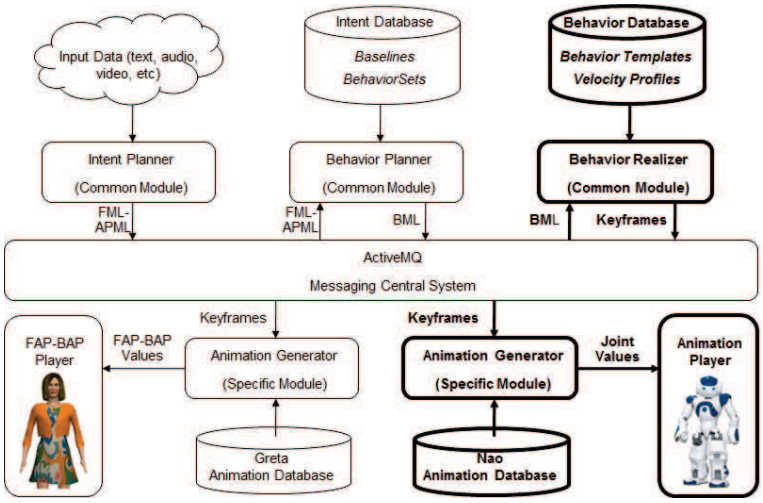


Figure 5.1: Our implemented modules in the new GRETA system. This work corresponds to the final stage of generation pipeline in the SAIBA framework: Synthesizing and executing multimodal behaviors

bolical descriptions for the gestures' shape to be reproduced; 2) Gesture Velocity Specification contains velocity profiles defining different speeds in a gesture action.

Different agents have the same procedures to handle their gesture database (see Chapter 4). However, as explained before, the data values defined in the database are agent-dependent because agents do not have the same movement capacities. Thus, one gesture database is created for the Greta virtual agent and another one is created for the Nao humanoid robot.

The following subsections describe each component of the gesture database in detail.

5.1.1 Gesture Repertoire

A gesture repertoire is also called a *Gestuary*. This name was first introduced by De Ruiter (1998) in his theoretical gesture model. Each entry in the gestuary is a pair of information: the name of the communicative intention and the description of the gesture shape which conveys the communicative intention. We use a similar definition of gestuary. The first part of each entry in the gestuary is used by the Behavior Planner module to select gestures to be realized. The second part is used by the Behavior Realizer module to instantiate selected gestures on the fly. For

instance, the entry described by the pairs (greeting, raising the right open hand over the head) represents a *greeting* gesture.

Requirements

An essential question arises when working with the gesture description: how to encode a human communicative gesture into a data unit so that our system can interpret and afterwards reproduce it expressively for a humanoid agent. It exists two main requirements for this task:

1. Human gesture should be encoded using enough information to rebuild this gesture without losing its signification. This means that the rebuilt gesture may not be fully identical with the original gesture but similar enough so as its original meaning is maintained to convey a predefined communicative intention. Hence, at least certain important elements that form a gesture have to be included in the description data.

2. The gesture description should stay at a high abstraction level so that the same syntax can be used to create gesture entries for different embodiments. It means that the description should not reference to specific animation parameters of agents (e.g., wrist joint). Hence, the gesture description is ought to be an extension of the existing higher-level behavior description language BML of the SAIBA framework. Following Heloir and Kipp (2010b), such a gesture description should be expressive, easy to use for the users and complete, precise and convenient to interpret for the animation engine (i.e. Behavior Realizer).

Behavior surface form and BML

The design of BML allows separating the surface form of a behavior and its instantiation in a communicative process (Kopp et al., 2006). It uses a set of XML elements and attributes to specify communicative verbal and nonverbal behaviors. The nonverbal behavior signals include posture, gesture, head, gaze, face. The standard BML language is used to exchange messages between the Behavior

Planner and the Behavior Realizer modules. Its specification focuses on the synchronization between behavior multimodal behavior signals and their feedback. In a BML message, the presence of behavior signals is indicated by their name taken out from their corresponding behavior lexicon. The detailed description of each signal's surface form is still open as mentioned in Vilhjálmsson et al. (2007) and Heloir and Kipp (2010b).

In the latest version of BML (i.e., version 1.0), a BML gesture signal is described by several attributes resumed in Table 5.1.

Standard Attribute	Type	Use	Description
id	ID	required	Unique ID makes reference to a particular gesture signal in the BML message
mode	closedSetItem	optional	Which hand is being used. Possible values: LEFT_HAND, RIGHT_HAND, BOTH_HAND
lexeme	openSetItem	required	An user-defined name that refers to a gesture template or a predefined animation
sync-points	sync attributes	optional	Seven standard sync-points are defined in Table 5.2 to specify absolute or relative timing values of a gesture signal

Table 5.1: Standard BML gesture behavior attributes

Sync Attribute	Description
start	beginning of gesture
ready	end of gesture preparation phase
strokeStart	start of the stroke
stroke	gesture stroke
strokeEnd	end of stroke
relax	start of retraction phase
end	end of gesture

Table 5.2: Synchronization attributes for a BML gesture

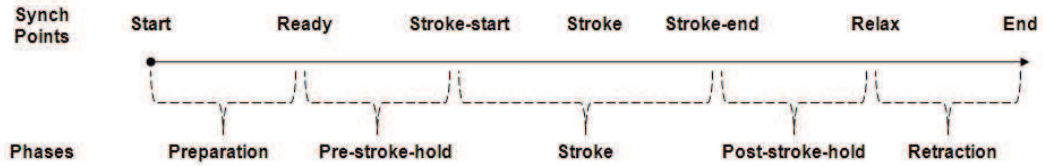


Figure 5.2: Standard BML synchronization points and gesture phases

The surface form of behavior signals is elaborated and stored in a behavior lexicon as illustrated Figure 5.3. Because the creation of such a surface form (e.g., gesture shape, facial expression, etc.) can be very time consuming, one of

objectives of the SAIBA framework is to have a common dictionary of behavior descriptions. This dictionary should be encoded in such a way that it can be shared between different SAIBA compatible systems. Thus, the SAIBA's researchers have expressed the need to develop the BML language to describe the surface form of signals in the behavior lexicon (Vilhjálmsson et al., 2007). Following this objective, in the next section we propose a gesture representation as an extension of the BML language to describe the gestures' surface form.

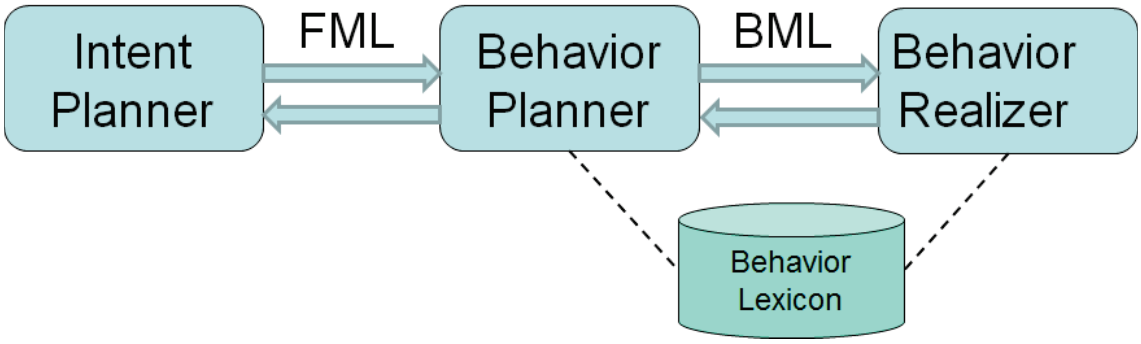


Figure 5.3: The SAIBA framework uses the BML language to communicate planned behaviors between Behavior Planner and Behavior Realizer. Additionally, SAIBA's researchers have planed to extend the BML language to describe the surface form of the behavior signals in the behavior lexicon (Vilhjálmsson et al., 2007).

Proposition

The set of standard BML attributes gives minimal information of a gesture surface form. This set needs to be extended by adding gesture properties to encode hand shapes and gesture phases. To do this, we rely on theoretical studies of gestures which are presented in Chapter 2. The gesture structure proposed by Kendon (2004) is used to describe the most important part of a gesture action and the description of gesture configuration by McNeill (1992) is used to specify gesture elements necessary to encode hand shapes. In detail, a gesture action may be divided into several phases of hand-arm movements, in which the main phase is called *stroke* transmitting the meaning of the gesture. The stroke phase may be preceded by a preparatory phase which serves to bring the articulatory joints (e.g.,

hand and arm) to a position where the stroke occurs. After that it may be followed by a retraction phase that returns the articulatory joints to a relax position or a position initialized for the next gesture.

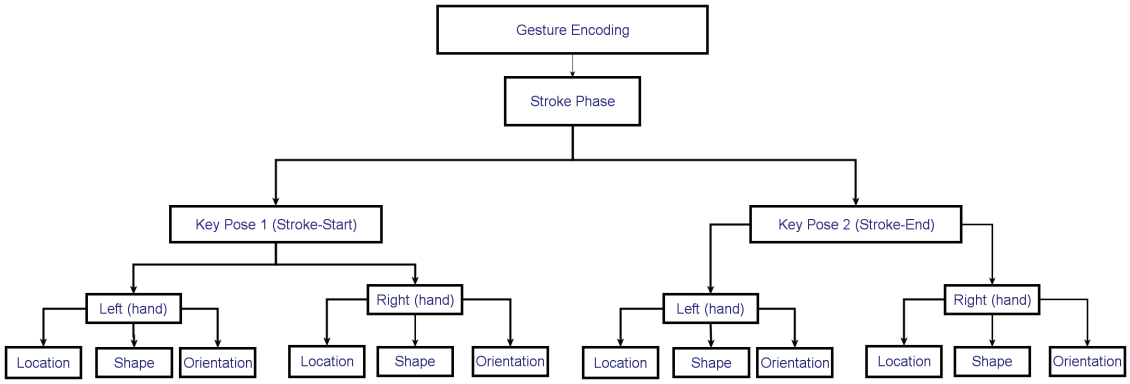


Figure 5.4: An example of gesture encoding: only the stroke phase is encoded. In this example the stroke phase has two key poses (i.e., Stroke-Start and Stroke-End) each of which is described with the information of each hand: hand shape, wrist position, palm orientation, etc.

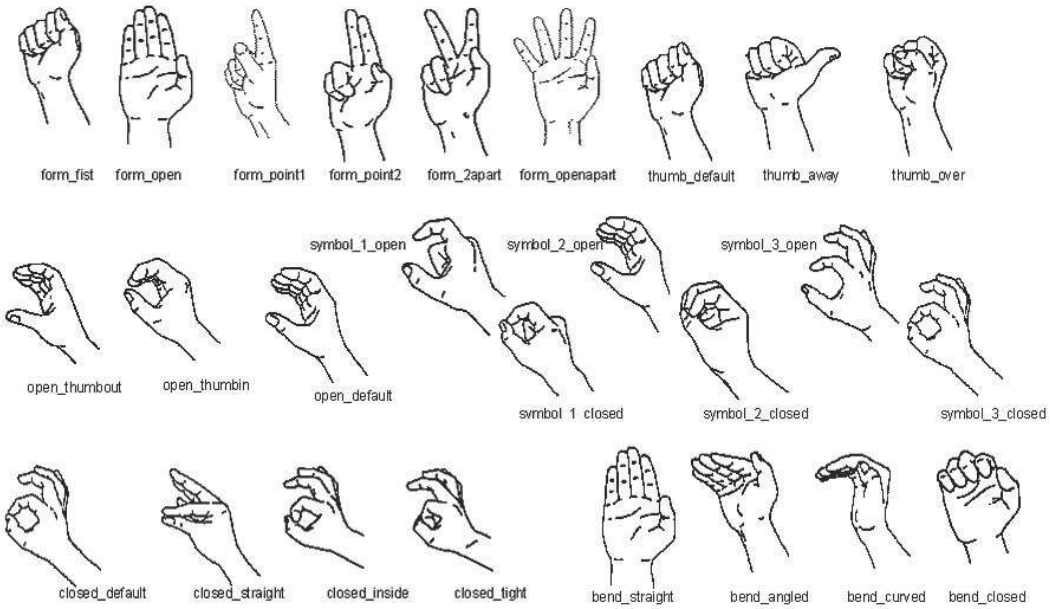


Figure 5.5: Some hand shapes from HamNoSys (Prillwitz, 1989)

To satisfy the first requirement (i.e., gesture should be encoded using enough

information to rebuild this gesture without losing its signification), in each entry of the gestuary (i.e., a gesture template), only the stroke phase is described. Other phases are generated automatically on the fly by our system. A stroke phase is represented through a sequence of key poses, each of which is described with information of both hands (left and right) such as their hand shape, wrist position, palm orientation as illustrated in Figure 5.4.

- *Hand Shape* is formed by variations on the configuration of the fingers (i.e., thumb, index, middle, ring and little). Figure 5.5 illustrates some hand shapes which are defined in the Hamburg Notation System (Prillwitz, 1989).

- *Wrist Location* This is the position of the hand wrist in a gesture space. The values of the wrist location are attributed according to its gesture space (e.g., McNeill's gesture space in Figure 2.4).

- *Palm Direction* The orientation of a hand is described by the orientation of the extended fingers and of the palm. Figure 5.6 illustrates some possible directions which are defined in the Hamburg Notation System (Prillwitz, 1989).

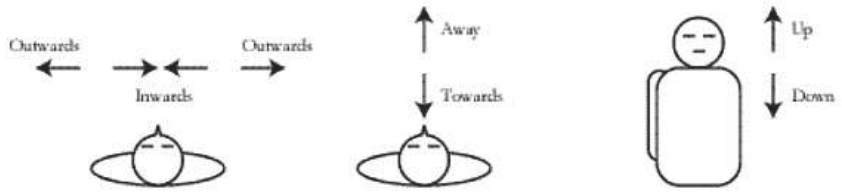


Figure 5.6: Directions defined for gestures from HamNoSys (Prillwitz, 1989)

- *Trajectory* is shaped by several basic movement primitives which are proposed like straight-line, curve, ellipse, wave, zigzag as illustrated in Figure 2.5.

Our objective is to develop a gesture representation which is an extension of the BML language (i.e., satisfying the second requirement of the gesture encoding task). Hence, we have proposed a BML compatible schema to encode gesture entries in a gestuary. A list of gesture attributes which are based on the HamNoSys

system (Prillwitz, 1989) is proposed to complete the gesture description as shown in Table 5.3

Extended Attribute	Description	Possible values
Category	The gesture category indicates the group to which the gesture belongs	Iconic, Metaphoric, Deictic, Beat
HandShape	The form made by a combination of configurations from Figure 5.5	form_open, thumb_away, etc
PalmDirection	The orientation of the palm from Figure 5.6	inwards, outwards, away, towards, up, down
ExtendedFingerDirection	The orientation of the extended fingers from Figure 5.6	inwards, outwards, away, towards, up, down
VerticalLocation	The vertical location of the wrist adopted from the concentric gestural space of McNeill (1992) as illustrated in Figure 2.4	YUpperEP, YUpperP, YUpperC, YCC, YLowerC, YLowerP, YLowerEP
HorizontalLocation	The horizontal location of the wrist adopted from the concentric gestural space of McNeill (1992) as illustrated in Figure 2.4	XEP, XP, XC, XCC, XOppC
FrontalLocation	The distance between the wrist and the breast of agent	ZNear, ZMiddle, ZFar
TrajectoryShape	The type of movement trajectory	Zic Zac, Circle, Straight, etc

Table 5.3: List of gesture attributes as an extension of BML

To keep the gesture description at an abstraction level, the possible values of the gesture attributes are symbolically setup. The symbolical values for wrist locations (vertical, horizontal, frontal) are adopted from the concentric gestural space of McNeill (1992) as illustrated in Figure 2.4. The abbreviations are used like *C* as Center, *CC* as Center-Center, *P* as Periphery, *EP* as Extreme Periphery, and *Opp* as Opposite Side.

Consequently, a gesture is encoded by specifying the key-poses of its stroke phase as shown in Figure 5.4 and attributes for each key-pose as declared in Table 5.3. This gesture encoding stored in a gestuary is formatted with an XML language. For each gesture phase, for each entry, different gesture forms are specified. Values for gesture elements that are not active in the given gesture, do not need to be specified.

In the example of Figure 5.7, the gesture involves the right arm: The stroke phase consists of two key poses (i.e., stroke-start and stroke-end). They are represented by information of the right hand positions (above the head), the hand shape (i.e., open hand) and the palm orientation (i.e., toward away). These two key poses are different by only one symbolic value of the horizontal position. This

gesture template specifies a hand movement that when repeated can be a greeting gesture.

```
1. <gesture lexeme="hello-waving"  mode="RIGHT_HAND">
2. <phase type="STROKE-START" trajectory="LINEAR" >
3.         <hand side="RIGHT">
4.             <vertical_location>YUpperPeriphery</vertical_location>
5.             <horizontal_location>XPeriphery</horizontal_location>
6.             <location_distance>ZNear</location_distance>
7.             <hand_shape>OPEN</handshape>
8.             <palm_orientation>AWAY</palm_orientation>
9.         </hand>
10. </phase>
11. <phase type="STROKE-END trajectory="LINEAR" >
12.         <hand side="RIGHT">
13.             <vertical_location>YUpperPeriphery</vertical_location>
14.             <horizontal_location>XExtremePeriphery</horizontal_location>
15.             <location_distance>ZNear</location_distance>
16.             <hand_shape>OPEN</handshape>
17.             <palm_orientation>AWAY</palm_orientation>
18.         </hand>
19.</phase>
20.</gesture>
```

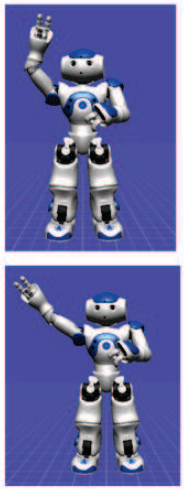


Figure 5.7: Example of a gesture template which specifies a hand movement that when repeated can be a greeting gesture.

Summary

We have developed a gesture representation language which is compatible with BML to encode gestures in a gestuary. This gesture representation language is defined at a high description level that is independent to a specific embodiment. Its specification allows describing gestures in an expressive way which is easy to elaborate by users and to interpret by a gesture engine.

5.1.2 Gesture Velocity Specification

In order to increase the credibility and the like-lifeness of a humanoid agent, its gestures should appear natural in their shape and speed. While the shape of gestures is specified by parameters presented in the previous subsection, characterizing speed of gesture movements is still a big challenge. To deal with this problem, we use the Fitts' law function (Fitts, 1992) to estimate the duration of linear hand movements in gesture trajectories.

Using Fitts' law to simulate human gesture speed

Fitts' law is an empirical model of human muscle movement that allows predicting the time necessary to move a hand or finger to reach a target rapidly (Fitts, 1992). This law has been concreted by several mathematical formulations. We followed the formulation used in human-computer interaction proposed by MacKenzie (1992). According to this formulation, the movement time MT is a function of the movement distance between the hand or finger and the reach point D and of the width of target W as described in Equation 5.1 below:

$$MT = a + b * log_2(\frac{D}{W} + 1) \tag{5.1}$$

where, a and b stand for the empirically adjustable parameters, *intercept* and *slope* respectively. Equation 5.1 forms a straight line in the plane. The slope b defines the gradient of that line and the intercept a defines the point at which the line crosses the y-axis, otherwise known as the y-intercept.

The index of difficulty ID to do a movement is described in Equation 5.2.

$$ID = log_2(\frac{D}{W} + 1) \tag{5.2}$$

and the index of performance IP is calculated in Equation 5.3.

$$IP = \frac{ID}{MT} \tag{5.3}$$

The difficulty with the Fitts' law is to find the a and b values. These indexes (ID, IP) will be used to find out the parameters a and b from a given training set of pairs (distance, time).

Some ECA systems such as (Kopp and Wachsmuth, 2002; Van Welbergen et al., 2005; Salem, 2012) also used the Fitts' law to simulate gesture movements in their gesture engine. They determined the parameters of Fitts' law with approximate values following a *trial and error* approach (i.e., a heuristic method to choose the best values after several attempts). For instance, Salem (2012) tested with the following 20 values for the slope coefficient $b = \{0.12, 0.14, 0.16, 0.18, 0.2, 0.22, 0.24, 0.26, 0.28, 0.3, 0.32, 0.34, 0.38, 0.4, 0.42, 0.44, 0.46, 0.48, 0.5\}$. After comparing the result for each of the 20 values in testing gesture trajectories with

the Honda robot (Hirai et al., 1998), value $b = 0.20$ was found to be the best choice. Similarly, in Kopp and Wachsmuth (2002)'s gesture engine for the MAX virtual agent, the intercept and slope coefficients (a,b) are set to (0;0.12) and the target width W has value 1 (i.e., W is neglected).

In our work, we extract these parameters from data of human movements. Thus, we proposed a reasonable procedure that includes two separate steps: i) retrieve data from real human gestures as input to train the Fitts' law function; ii) build a regression line equation to find out corresponding parameters. The following paragraphs describe in detail our procedure:

Retrieve training data We collected a set of communicative gestures from analyzing a storytelling video corpus made by Martin (2009). The spatial and temporal information of phases in a gesture was annotated using the Anvil tool (Kipp et al., 2008) as illustrated in Figure 5.8. The extracted data correspond to 3D human movement on a 2D video screen. We are aware that these data are an approximation of real data that could have been acquired through motion capture data. However, we did not have such an equipment at our disposal.

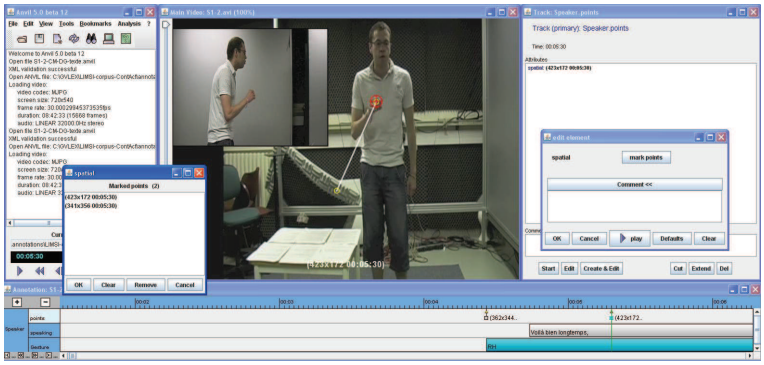


Figure 5.8: Spatial annotation with the Anvil tool

Each annotated phase in a gesture action gives two pairs of information including the wrist trajectory (i.e., start and end positions during each gesture phase) and the movement duration. We assumed that the small path in a gesture trajectory is linear so that it is enough to compute its distance using only start and end positions. As a result, we can calculate how far and for how long a gesture was

realized. Twenty five gesture phases were annotated. They were collected from three different actors filmed in the video corpus.

Build regression line equation From gesture annotations obtained in the first step (i.e., duration and distance of gesture trajectories), we applied Equations 5.1 and 5.2 and 5.3 to establish a table (i.e. Table 5.4). Its first column is the value of the difficulty of a gesture movement (ID). Then the second and third columns correspond to the distance and the duration of gesture trajectories. In the last column of the table the index of performance IP was calculated.

ID	D	MT (ms)	IP
1.36	51.36	600	2.27
-0.25	16.88	200	-1.22
1.06	41.84	470	2.27
0.23	23.58	260	0.91
0.53	29.03	370	1.45
1.79	69.37	1330	1.35
1.18	45.59	500	2.38
1.14	44.30	500	2.29
1.78	68.77	530	3.36
0.27	24.25	360	0.75
0.32	68.48	430	0.74
0.73	9.24	630	1.15
1.06	25.69	630	1.68
0.86	30.98	570	1.50
0.46	15.80	400	1.15
0.71	42.94	660	1.07
1.04	35.62	600	1.73
0.32	15.76	530	0.60
-0.32	37.32	300	-1.06
1.58	81.16	870	1.81
1.54	49.05	930	1.65
0.56	23.26	200	2.80
-0.17	17.63	360	-0.47
1.09	8.83	670	1.62
1.24	40.53	430	2.88

Table 5.4: Retrieved data from real humans

A Fitts' law regression line equation was built for this data as described in Equation 5.4. The points of the coordinates (MT, ID) are drawn on a 2D space. Then we computed the regression line approximating these points and get turned the a and b values when a is the slope of the line and b is the intercept.

$$MT = 292.9 + 296.6 * log_2(\frac{D}{W} + 1) \quad with \quad R^2 = 0.532 \tag{5.4}$$

where $a = 292.9$ and $b = 296.6$. The correlation coefficient R^2 describes how

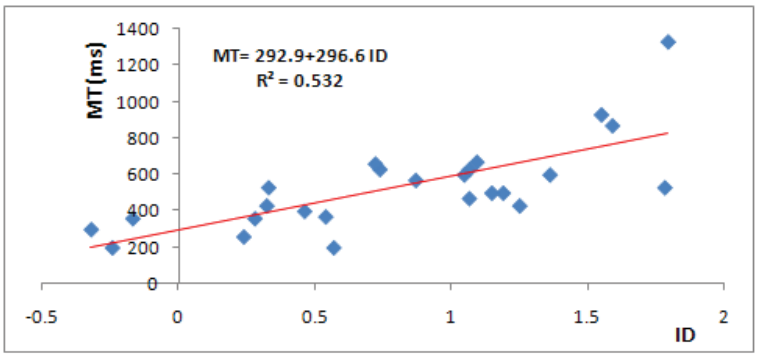


Figure 5.9: Scatter plot and regression line for data in Table 5.4

well the fit of a regression line equation is built from a given data. The closer its value is to 1.00, the better the fit is. That means if its value is 1.00, all points must belong to the regression line. In our experiment, the correlation coefficient's value is not high (i.e., $R^2 = 0.532$) because there are few points of coordinates (MT, ID). However, its value is greater than 0.5 so that the result is enough to be confident.

Limited speed of gesture movement

The robot has a limited speed in its body movement. As a result, in some cases, the speed computed by the Fitts' law could be greater than the maximal speed of the robot. In these cases, our system has to use the maximal speed to calculate the duration of a robot gesture. This information is required by the gesture scheduler module to find out if there is enough time to perform a gesture; that is we have to know what is the maximal speed with which a gesture can be executed.

Position (from-to)	keypos1	keypos2	keypos3	keypos4	...
keypos1	0.0	0.18388	0.28679	0.2270	...
keypos2	0.18388	0.0	0.19552	0.2754	...
keypos3	0.28679	0.19552	0.0	0.3501	...
keypos4	0.2270	0.2754	0.3501	0.0	...
...	...	...	...	...	...

Table 5.5: Minimum durations necessary to do a hand-arm movements between any two positions (coded as keypos1, keypos2, etc) in the Nao gesture space

We define a robot gesture space including all positions a robot hand can reach

when gesturing. These positions have symbolical values such as {(XCC, YUpperC, ZNear), (XP, YLowerP, ZMiddle), etc} as described in Table 5.3. Each symbolical position corresponds to a combination of the robot's joint values as shown in Table 5.6.

Code	X	Y	Z	Values (ShoulderPitch, ShoulderRoll, ElbowYaw, ElbowRoll)
keypos1	XEP	YUpperEP	ZNear	(-54.4953, 22.4979, -79.0171, -5.53477)
keypos2	XEP	YUpperEP	ZMiddle	(-65.5696, 22.0584, -78.7534, -8.52309)
keypos3	XEP	YUpperEP	ZFar	(-79.2807, 22.0584, -78.6655,-8.4352)
keypos4	XEP	YUpperP	ZNear	(-21.0964, 24.2557, -79.4565, -26.8046)
...	...	...	...	...

Table 5.6: Gesture Space Specification for the Nao robot

These symbolical positions are called key positions and are abbreviated as keypos1, keypos2, etc in Table 5.5. In this table, the minimum time (i.e., maximal speed) necessary to do a hand-arm movement between any two key positions in the robot gesture movement space is given. To fill in this table, we recorded the average time that Nao took to do each movement at its maximal speed. That is intrinsic to each robot joint. For this purpose, we developed an algorithm based on available APIs from NAO SDK (Gouaillier et al., 2009, 2008). We notice that the values in Table 5.5 are robot-dependent.

As a result, the same Fitts' law equation in 5.4 is used to calculate average movement time of a gesture action for both agents, virtual and physical agents, but the minimum time within which the gesture can be realized is different for each agent (the values in Table 5.5 are for the Nao robot).

Summary

We have used the Fitts' law function to simulate the timing of human gestures in our model. The parameters of this function and the agent-dependent minimum durations tables are stored as external parameters of the system. They are useful to eliminate gestures for which the allocated time is less than the necessary time to create the gestures.

5.2 Behavior Realizer

The Behavior Realizer module is one of the four main modules in the GRETA system as illustrated in Figure 5.10. This module receives as input BML messages from the Behavior Planner module via the messaging central system ActiveMQ (Snyder et al., 2011) in realtime. After processing multimodal behaviors specified in BML messages, this module returns a set of keyframes to the ActiveMQ's network.

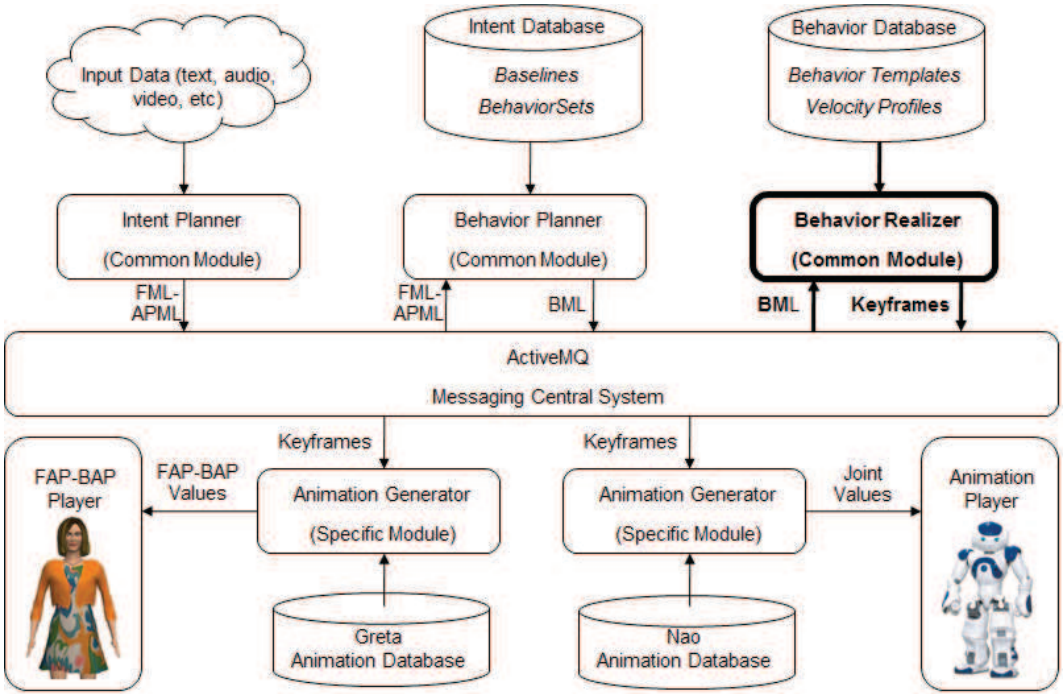


Figure 5.10: The Behavior Realizer module is integrated within the GRETA system

The processes in the Behavior Realizer module is common to all types of agents. It means that they are independent of the agent's embodiment and the animation player technology. An agent-dependent behavior database is elaborated as an external parameter (see previous section 5.1).

In brief, there are three computational stages in this module. Firstly, the system validates the received BML messages and instantiates the BML tags of gesture signals from gesture templates to rebuild complete gestures. Secondly,

these gestures are scheduled to be synchronized with speech while taking into account the gesture velocity specification and expressivity parameters. Thirdly, the module generates a set of gesture keyframes in which each keyframe contains a description of each gesture phase and timing information (i.e., absolute values). The data flow from BML to keyframes is illustrated in Figure 5.11.

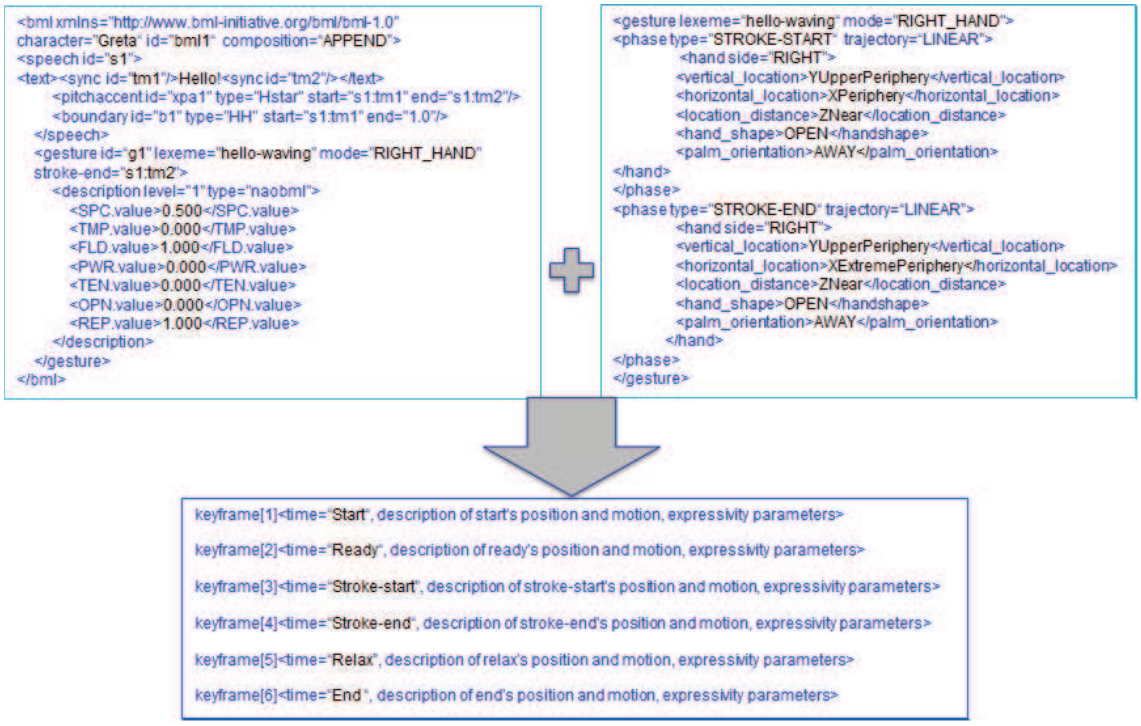


Figure 5.11: From BML and Repository to Keyframes

5.2.1 BML Resolver

The GRETA system follows the SAIBA framework. Hence we use the syntax defined by the standard version of Behavior Markup Language (Website, 2011) to encode behavior scripts generated by the Behavior Planner module and received by the Behavior Realizer module. Our BML Resolver submodule is developed to deal with the BML semantic descriptions. This submodule interprets behaviors specified in each BML request which it receives. Three issues need to be addressed in our implementation of the BML Resolver module: 1) Initializing behavior tim-

ing; 2) Composition of BML messages; 3) Rebuilding the behavior surface form. The following subsections will present each issue in detail.

Initializing behavior timing

Multimodal behaviors may occur either simultaneously or sequentially. The system has to interpret temporal dependencies between the behaviors for two objectives: 1) detect a temporal conflict which leads to an infinite loop (i.e., a state of waiting forever) between signals as illustrated in Listing 5.1; 2) initialize the timing for each behavior signal.

To recall, a BML message is a set of communicative behaviors within a <bml> element. Each BML behavior has several synchronization points (i.e., sync-points such as start, ready, stroke, etc) which can be described relative to the sync points of other behaviors.

Listing 5.1: Temporal conflict between BML signals: **s1.start** = p1.end = p1.start + 2 = g1.stroke + 2 = **s1.start + 2 = ...**

```
<bml xmlns="http://www.bml-initiative.org/bml/bml-1.0" id="bml1" characterId="
    Greta" composition="APPEND">
    <speech id="s1" start="p1:end"
        <text> I don't think so!</text>
    </speech>
    <posture id="p1" lexeme="STAND" start="g1:stroke" end="start+2"/>
    <gesture id="g1" stroke="s1:start" lexeme="DENY"/>
</bml>
```

The BML message in Listing 5.1 has an infinite loop. The speech starts at the end of the posture signal. The posture signal, in turn, starts at the same time as the stroke of the gesture signal whose timing is relative to the speech's starting.

We have implemented a non-exhaustive algorithm to detect possible conflicts between behaviors specified in a BML message. The idea is as follows: we define a path (sync-point1, sync-point2, sync-point3,...) as a relative timing path in which sync-point1 is relative to sync-point2, sync-point2 is relative to sync-point3, etc. For instance, one found path in Listing 5.1 is (s1.start, p1.end, p1.start, g1.stroke, s1.start, ...). If we can find out a relative timing trajectory whose length (i.e., number of sync points included in the path) is more than the number of all sync

points in the BML message, then there exists an infinite loop which is detected by our algorithm illustrated in Figure 5.12.

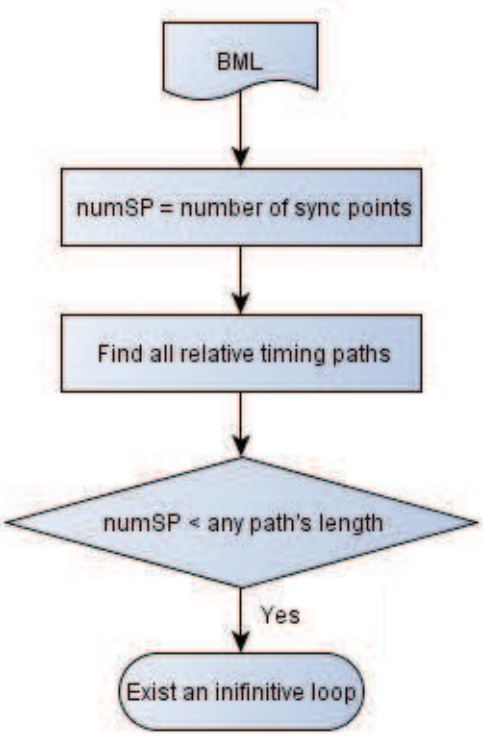


Figure 5.12: Check the existing of an infinite loop

Taking the BML message in Listing 5.1 as example to illustrate our algorithm: the number of sync points declared in the BML message is 4 including (s1.start, p1.end, p1.start, g1.stroke), and we have a relative timing path (s1.start = p1.end = p1.start+2 = g1.stroke+2 = s1.start+2 = ...) whose length is greater than 4. Thus, there exists an infinite loop in this BML message.

After this process, if an infinite loop exists, the whole BML message is canceled. Otherwise, relative timing points of given behaviors are instantiated. If we consider the BML example shown in Figure 5.11, the stroke timing point is initialized in relation with the speech timing. The timing information of other gesture phases such as preparation and relaxation phases are calculated from this stroke timing. These different temporal values can be used to synchronize behaviors of other modalities with this gesture. This synchronization computation procedure

is presented in detail later.

Composition of BML messages

If a new BML request arrives before the realization of previous requests has been completed, a BML composition mechanism must be developed to resolve them. The BML standard defines three cases for behaviors composition (i.e., via *composition* tag):

1. The behaviors specified in the new BML block and in the previous BML blocks are computed and realized together as one BML request. If a conflict exists, the new behaviors cannot modify behaviors defined in a previous BML requests (i.e. merging composition).

2. The new BML block has to wait for the prior blocks to end before starting the new one (i.e., appending composition). This means that all signals in the current BML have to finish and return to rest states before any of the signals in the new BML starts.

3. The behaviors specified in prior BML blocks are forced to stop when the new BML block arrives. In the later case, the behaviors defined in the new block are realized as usual (i.e., replacing composition).

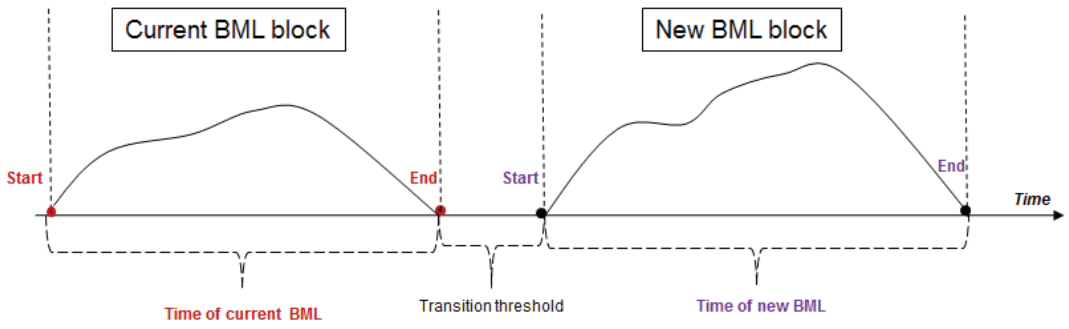


Figure 5.13: Appending composition of two BML messages

So far, only the appending composition was implemented in our system as illustrated in Figure 5.13. The new coming BML message is separated from the

current BML message by a temporal transition threshold. This means that the execution of all signals in a BML message has to be completed before starting a new one. The start and end timing of a BML message are calculated by the start timing of its earliest signal and the end timing of its latest signal.

The other cases of replacing and merging composition have not yet been integrated. These cases require modeling of the interruption and the combination of gestures which are being executed.

Fill up behavior surface form

As indicated in the previous section (i.e., database section), the behavior surface form is elaborated and stored in a repository. When a behavior signal is specified in a BML message, its shape is rebuilt from its corresponding symbolic template in its repository. Figure 5.14 illustrates the shape of a hand gesture indicated in the BML message filled up from a gestuary via *lexeme* tag.

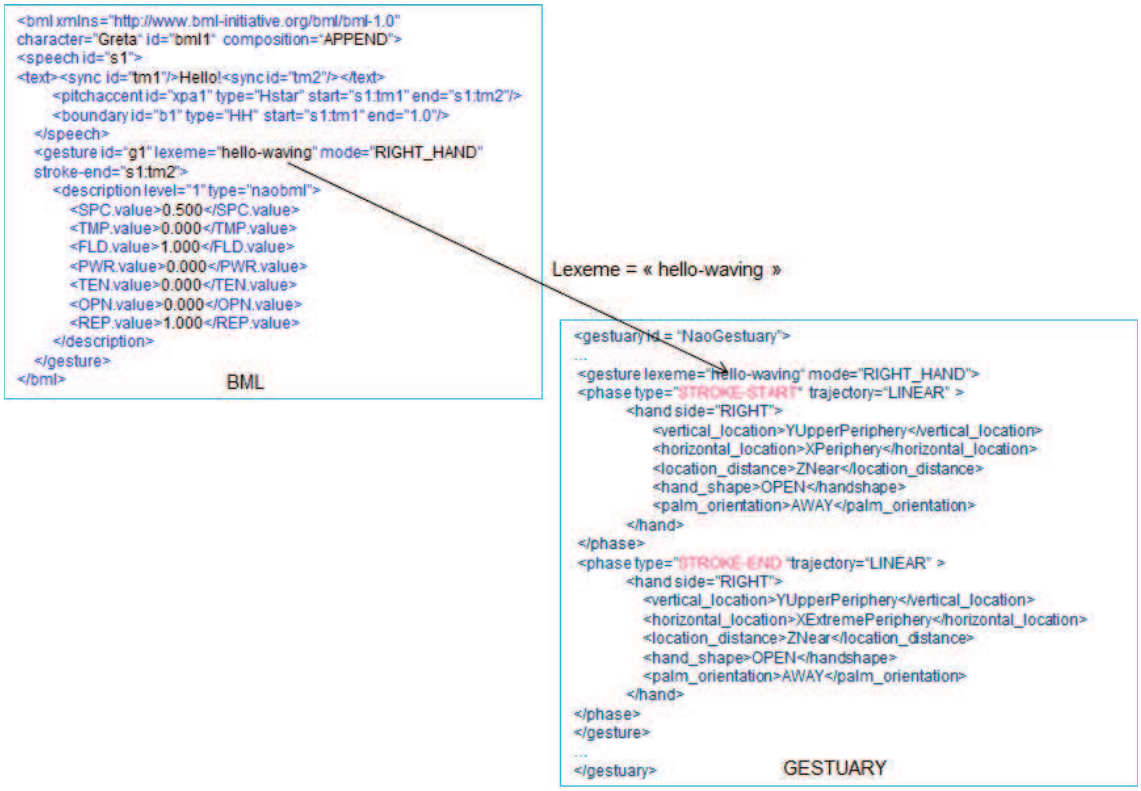


Figure 5.14: Fill up gesture surface form

The output of this process is a list of behavior signals whose surface form is completed (e.g., the description of gesture phases).

5.2.2 Gesture Scheduling

One main requirement to schedule gestures is that gestures are tightly coordinated with speech. In our system, the synchronization between gesture signals and speech is ensured by adapting the timing of gestures to speech timing. It means that the temporal information of gestures within a *BML* tag are relative to speech.

We have implemented an algorithm to schedule gestures following three stages. In the first stage, the duration to execute each gesture phase (i.e., preparation, stroke, retraction) of one gesture is pre-calculated. This information is necessary to ensure that the gesture starts at the right time so that the stroke happens on the stressed syllables of the speech (i.e., the condition to synchronize gesture and speech). In the second stage, the start and end timings of phases of each gesture are instantiated using the speech timing. This means that the timing of the gesture's synchronization points (i.e., sync points) is instantiated in real values. In the third step, we handle all of the planned gestures together to create gesture trajectories. The issues of co-articulation between consecutive gestures will be handled in this step.

STAGE 1

The duration to perform a gesture trajectory from one position to the next position in a natural speed is calculated using the Fitts' law in Equation 5.4. However, for the Nao robot, the duration computed by the Fitts' law may be shorter than the duration necessary (i.e., minimum duration) to execute the gesture trajectory. In this case, the system has to use the minimum duration instead of the Fitts' law duration as illustrated in Figure 5.15.

For each gesture, three durations are computed for its phases: 1) the preparation phase (i.e., hand moves from a rest position (RE) to the *stroke-start* (SS) position); 2) the stroke phase (i.e., hand moves from *stroke-start* (SS) to *stroke-end* (SE) positions); and 3) the retraction phase (i.e., hand move from *stroke-end* (SE) position to a rest position (RE)) respectively. The durations of the *pre-stroke-hold*

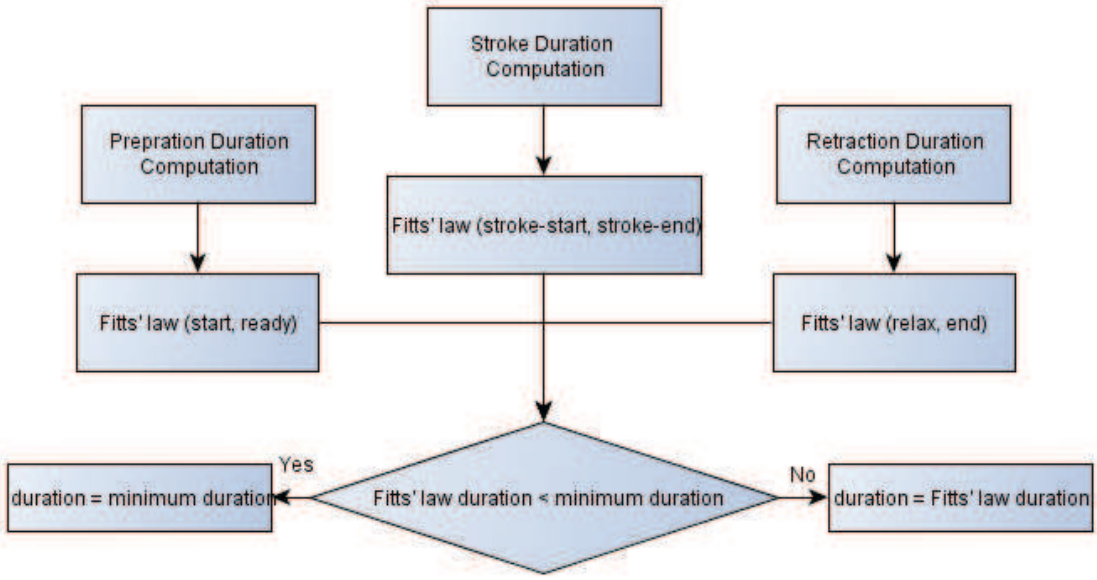


Figure 5.15: Compute gesture phases' duration

and the *post-stroke-hold* phases are relative to the whole gesture's duration and are calculated following Equation 5.5.

$$
\begin{aligned}
&strokeDu = Fitts(SS, SE) > MinD(SS, SE)?Fitts(SS, SE) : MinD(SS, SE)\\
&preDu = Fitts(RE, SS) > MinD(RE, SS)?Fitts(RE, SS) : MinD(RE, SS)\\
&retDu = Fitts(SE, RE) > MinD(SE, RE)?Fitts(SE, RE) : MinD(SE, RE)\\
&preStrokeHold = (strokeDu + preDu + retDu) * ratioPreHold\\
&postStrokeHold = (strokeDu + preDu + retDu) * ratioPostHold
\end{aligned}
\tag{5.5}
$$

In Equation 5.5, by default the parameters *ratioPreHold* and *ratioPostHold* are experimentally set to 0.3 and 0.6 respectively. However, the values of *pre-stroke-hold* and the *post-stroke-hold* phases are not fixed but are further modulated to be in line with the time allocated to the gesture (see below).

The result of this stage is five real values for the durations: stroke phase duration, preparation phase duration, retraction phase duration, pre-stroke-hold phase

duration and post-stroke-hold phase duration. These values are named *preparationDuration*, *strokeDuration*, *retractionDuration*, *preStrokeHold*, *postStrokeHold* respectively and will be used in the next stage.

STAGE 2

The objective of this stage is to calculate the timing value for sync points of a gesture. Some of the sync points may be concretized with real values directly from the speech time markers through their relative timing. Other sync points which are not yet concretized have to be computed in this stage. There are several cases to be addressed:

Case 2.1: The *stroke-end* sync point is specified in the BML message as illustrated in Listing 5.2.

Listing 5.2: *stroke-end* is first instantiated from speech timing

```
<bml>
    <speech id="s1" start="1"> <text>I don't <sync id="tm1"/> think so! </text>
    </speech>
    <gesture id = "g1" strokeEnd = "s1:tm1" mode = "BOTH_HAND" lexeme="DENY"\>
</bml>
```

The timing of the stroke phase is the most important information in order to synchronize the gesture with the speech. The timing of the stroke-end sync point coincides with emphasized word(s) in speech. Thus, if the stroke timing (i.e., *stroke-end*) is declared in the BML message, it has to be first instantiated with real value via the gesture's time markers relative to the speech timing. Other sync points such as *stroke-start*, *ready*, *relax*, *start*, *end* are calculated from the value of *stroke-end* following Equation 5.6 and illustrated in Figure 5.16:

$$\begin{aligned} strokeStart &= strokeEnd - strokeDuration \\ ready &= strokeStart - preStrokeHold \\ start &= ready - preparationDuration \\ relax &= strokeEnd + postStrokeHold \\ end &= relax + retractionDuration \end{aligned} \tag{5.6}$$

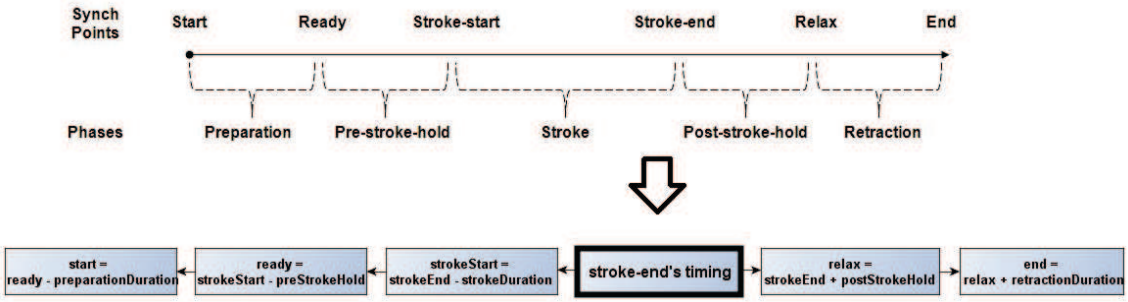


Figure 5.16: Other sync points are computed from the value of stroke-end

where *strokeDuration, preStrokeHold, preparationDuration, postStrokeHold, retractionDuration* are computed following Equation 5.5.

In case the stroke-start is also declared in the BML message as illustrated in Listing 5.3. Then the allocated time is constrained within the stroke-start and stroke-end sync points. First of all, a condition has to be computed to verify whether the necessary time is less than the allocated time to execute the stroke phase. The whole gesture will be deleted if there is not enough time to execute the stroke phase. Otherwise, the value of stroke-start is recalculated following Equation 5.6.

Listing 5.3: Both *stroke-start* and *stroke-end* are declared in the BML message

```
<bml>
    <speech id="s1" start="1"> <text> <sync id="tm1"/> I don't <sync id="tm2"/>
        think so! </text>
    </speech>
    <gesture id = "g1" strokeStart = "s1:tm1" strokeEnd = "s1:tm2" mode = "
        BOTH_HAND" lexeme="DENY"\>
</bml>
```

Case 2.2: The stroke timing is not declared but the end and start sync points are concretized in the BML message as illustrated in Listing 5.4.

Listing 5.4: the start and end time markers are declared in the BML message

```
<bml>
    <speech id="s1" start="1"> <text> <sync id="tm1"/> I don't think so! <sync
        id="tm2"/> </text>
    </speech>
```

```
    <gesture id = "g1" start = "s1:tm1" end = "s1:tm2" mode = "BOTH_HAND" lexeme
        ="DENY"\>
</bml>
```

In this case, we compute other sync points' time (i.e., ready, stroke-start, stroke-end, relax) from these given start's value and end's value following Equation 5.7.

$$\begin{aligned} ready &= start + preparationDuration \\ strokeStart &= ready + preStrokeHold \\ strokeEnd &= strokeStart + strokeDuration \\ relax &= end - retractionDuration \end{aligned} \tag{5.7}$$

where *strokeDuration, preStrokeHold, preparationDuration, postStrokeHold, retractionDuration* are computed following Equation 5.5.

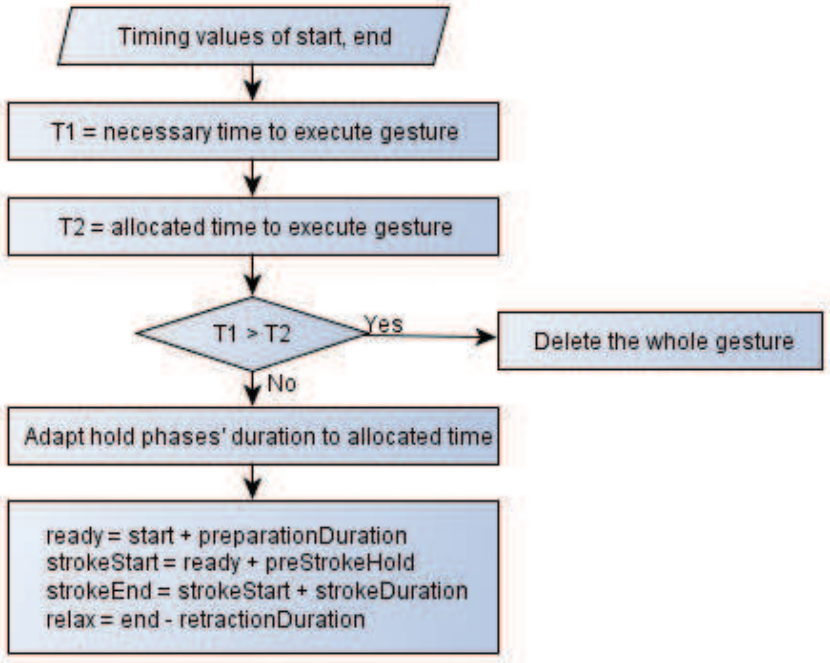


Figure 5.17: Other sync points are computed from the values of start and end

We need to check whether the allocated time (constrained within start and end sync points) is enough to do the gesture. It means that the allocated time must not

be less than the minimum time necessary to execute the gesture. The minimum time necessary is calculated by summing the duration of three phases in a gesture: preparation, stroke and retraction phases. If this condition is not satisfied, the whole gesture is deleted. Otherwise, we continue the instantiation of other sync points as illustrated in Figure 5.17. In this algorithm, the timing of hold phases is dynamically planned. For instance, if the allocated time is much more than the necessary time, the hand gesture may move slowly. To avoid this, the hold duration after the stroke phase must be increased by setting the relax sync point later following Equation 5.7 (i.e., the duration of the hold phase is calculated by (*relax-strokeEnd*) in which the *strokeEnd* is not changed). Otherwise, the duration of hold phases is decreased to fit the allocated duration.

Case 2.3: No sync points are declared for the gesture in the BML message as illustrated in Listing 5.5.

Listing 5.5: No sync point is declared for the gesture

```
<bml>
    <speech id="s1" start="1"> <text> I don't think so! </text>
    </speech>
    <gesture id="g1" mode = "BOTH_HAND" lexeme="DENY"\>
</bml>
```

Following the definition of the BML language, *"If no timing constraints are given, the behaviors are all expected to start immediately and run for their default durations."* (Kopp et al., 2004b). That means, the gesture and the speech in this case start at the same time. The start of the gesture is set to zero and then other sync points are calculated based on the gesture's default durations following Equation 5.8.

$$
\begin{aligned}
start &= 0 \\
ready &= start + preparationDuration \\
strokeStart &= ready + preStrokeHold \\
strokeEnd &= strokeStart + strokeDuration \\
relax &= strokeEnd + postStrokeHold \\
end &= relax + retractionDuration
\end{aligned}
\tag{5.8}
$$

where *strokeDuration, preStrokeHold, preparationDuration, postStrokeHold, retractionDuration* are computed following Equation 5.5.

This case does not ensure the synchronization of gestures and speech. But it allows that our Behavior Realizer runs as a BML Realier as it deals with all possible cases of a BML message in the SAIBA framework.

STAGE 3

The objective of this step is to compute how two consecutive gestures are co-articulated. For each gesture in the sequence, there exists three cases to be addressed:

Case 3.1: The current gesture and its next gesture are separated (i.e., their sync points do not coincide). This means that the current gesture has finished before the next gesture to start. This case is illustrated in Figure 5.18.

If the distance in time d between two gestures (i.e., d is set to the distance in time between the end of the current gesture and the start of the next gesture) is less than a gestural transition threshold (i.e., $d < threshold$), there will be a co-articulation from the current gesture to the next gesture as Case 3.3 below. Otherwise, the current gesture goes to a rest position.

Case 3.2: The current gesture and its next gesture overlap: the current gesture's stroke has not yet finished when the next gesture's stroke starts as illustrated in Figure 5.19. Thus, the next gesture will be deleted.

Case 3.3: Similarly to the case 2, the current gesture and the next gesture overlap. But the current gesture's stroke has already finished when the next ges-

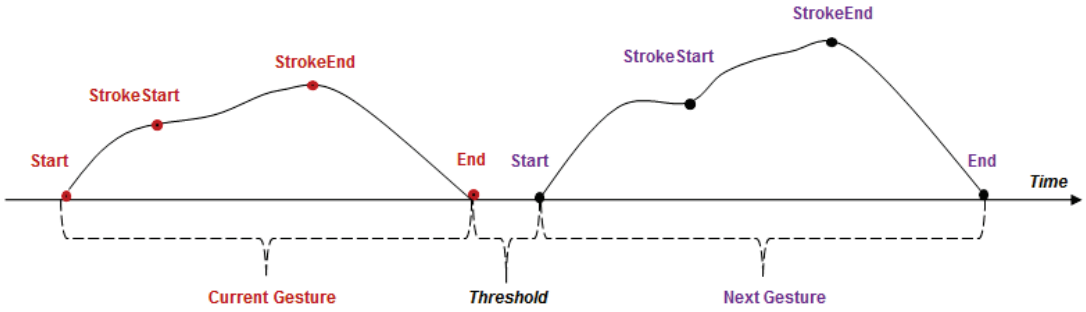


Figure 5.18: No co-articulation between two gestures

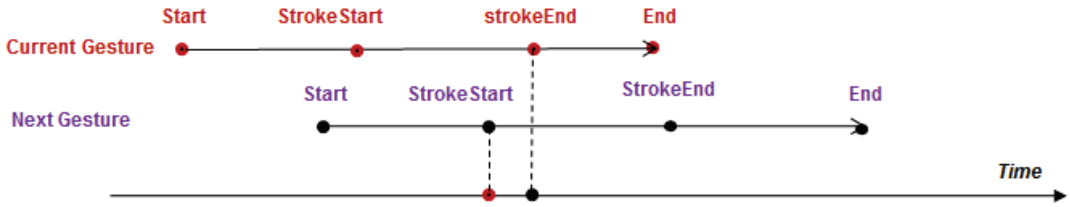


Figure 5.19: Two gestures are overlapped.

ture's stroke starts as illustrated in Figure 5.20.

This is a case of co-articulation: the hand moves from the stroke-end's position of the current gesture to the stroke-start's position of the next gesture without going through a relax position. We have to estimate the time necessary to do this movement. If this duration is too long so that the hand will not reach the stroke-start's position of the next gesture in the planned time, the next gesture must be deleted as speech-gesture synchronization cannot be guaranteed. Otherwise, a co-articulation between them is computed: 1) The retraction phase of the current gesture will be canceled; 2) The next gesture will start from the stroke end position of the current gesture instead of its start position; 3) The time to start the next gesture's preparation phase is reset to adapt to its current start position.

The solution for this case is described in Figure 5.21. The process of the right hand is separated from the process of the left hand. Thus, the algorithm is repeated twice for each hand side.

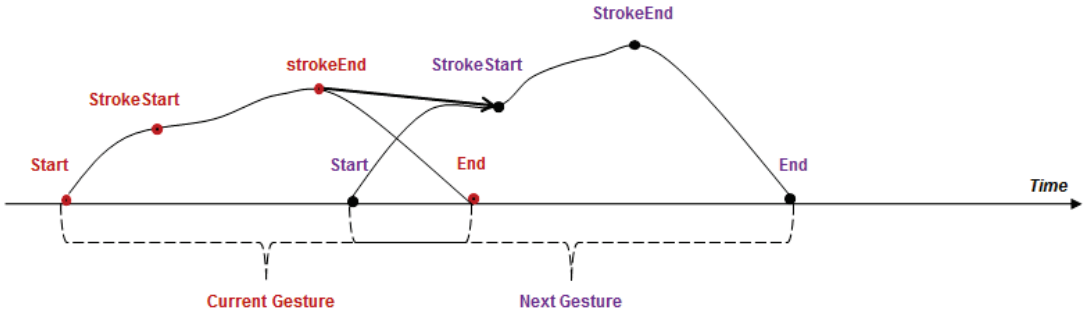


Figure 5.20: Two gestures overlap and the hand moves from the stroke-end's position of the current gesture to the stroke-start's position of the next gesture.

5.2.3 Gesture Expressivity

Regarding gesture expressivity, it is specified by a the set of expressivity parameters described in BML for each gesture signal. This set is divided into two subsets. The first subset including spatial extent (SPC), temporal extent (TMP) and stroke repetition (REP) is taken into account whilst the timing of gesture phases is calculated. The second subset including other parameters of the set (i.e., fluidity, power, openness, tension of gesture movements) is applied when creating the gesture animation. The reason for this is that the expressivity parameters in the second subset is dependent on an agent's embodiment. For instance, the Nao robot does not support the acceleration modulation of the gesture movements in realtime. We now describe how we implement the parameters of the first subset.

Temporal Extent - TMP

The TMP parameter modifies the duration of the entire gesture (i.e., including the duration of preparation, stroke and retraction phases) from the slowest speed (i.e. when the value of this parameter is set to -1) to the fastest speed (i.e., when the value of this parameter is set to 1). Hence the values of this parameter are between [-1,1] in which the value of zero corresponds to a *neutral* state. Following Hartmann et al. (2006) this *neutral* state is not related to the neutral state of human gestures, but it refers to a state where the behavior is realized without any expressivity in our model.

If the TMP value increases, the duration of the gesture decreases. This means

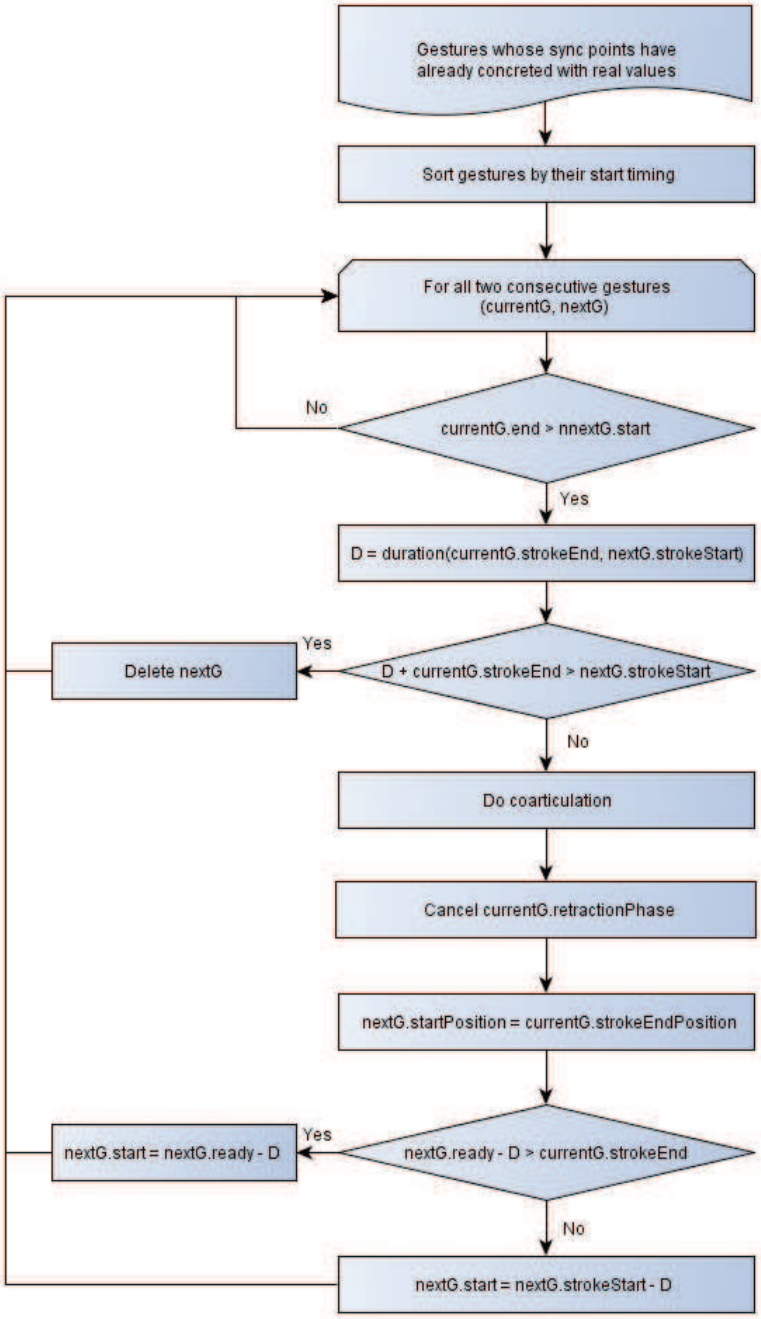


Figure 5.21: Algorithm to compute how two consecutive gestures are co-articulated

that the duration of each gesture phase decreases. As studied in the previous section (i.e., Gesture Database), we have already two duration values for each

phase: the minimum duration (i.e., in maximal speed) and the natural duration (i.e., following the Fitts' law equation). Thus, the duration of a gesture phase should not be less than its minimum duration. As a result, we can formulate an equation in which if TMP is set to 1, the duration is equal to the maximal duration, and when TMP is set to 0, the duration is equal to the natural duration as described in Equation 5.9.

$$duration = fittsDuration + (minDuration - fittsDuration) * TMP \quad (5.9)$$

In order to maintain the synchronization between gesture and speech, the time of the *stroke-end* sync point is not changed. Consequentially, the start, ready and stroke-start sync points are later and the relax and end sync points (if any) are earlier following Equation 5.6. This performance is illustrated in Figure 5.22.

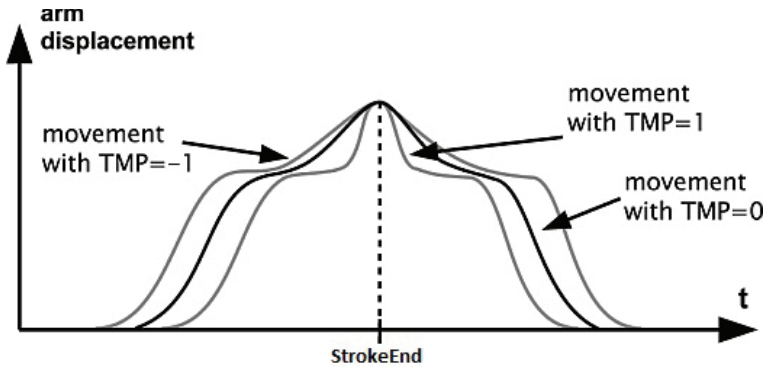


Figure 5.22: The TMP varies on execution of a gesture (Hartmann et al., 2006)

Spatial Extent - SPC

The SPC parameter affects the amplitude of wrist movements from very small and contracted (i.e., when the value of this parameter is -1) to very large and expanded (i.e., when the value of this parameter is 1) in a gesture space as illustrated in Figure 5.23. Similarly to the TMP parameter, the values of this parameter vary between [-1,1] in which the value of zero corresponds to a *neutral* state where agent behaviors are realized without any expressivity (Buisine et al., 2006). Now let's

take a case of increasing the SPC value to illustrate the effect of this parameter to modulate gesture animation.

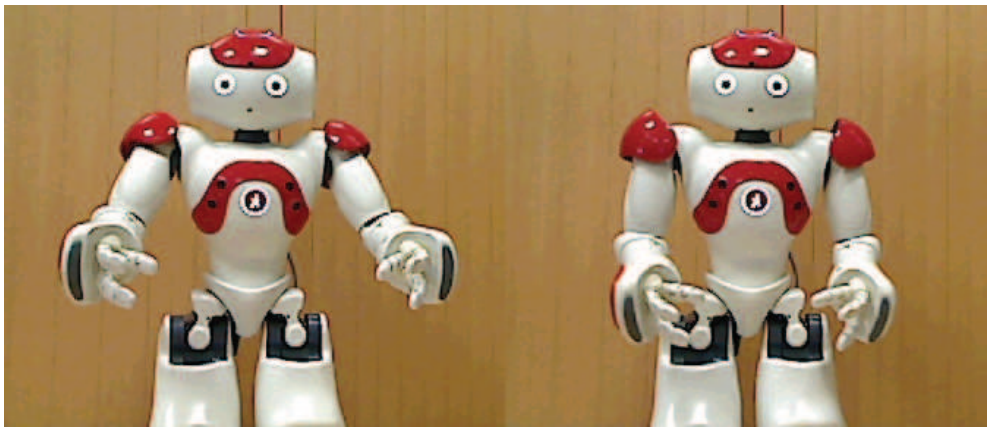


Figure 5.23: The SPC affects on the execution of an iconic gesture (left: SPC = 0; right: SPC = -1)

When the SPC value increases, the locations of the wrists of current gesture are higher and further away from the agent's body. As a result, three values (*vertical*, *horizontal*, *distance*) of the wrist position are increased. This means that the trajectory distance of the gesture changes.

For the Nao humanoid robot, due to its gesture space limitation (i.e., it has several singular positions), positions where the robot wrist can reach are predefined, called key-positions. In Section 5.1 we defined the robot gesture space which is based on McNeill's gesture space. This robot gesture space has 5 horizontal values (i.e., XOppC, XCC, XC, XP, XEP), 7 vertical values (i.e., YLowerEP, YLowerP, YLowerC, YCC, YUpperC, YUpperP, YUpperEP) and 3 distance values (i.e., ZNear, ZMiddle, ZFar) as shown in Table 5.3. As a result of combining these values, there are 105 key-positions of the robot wrist whose joint values are predefined ahead of time.

Thus, the effect of the SPC value on the robot gestures is constrained within the predefined 105 key-positions. So far, there are only three values of the SPC parameter defined at -1, 0 and 1. This means that, after applying the SPC parameter, the position of the robot wrist (X[i],Y[j],Z[k]) becomes (X[i+SPC], Y[j+SPC], Z[k+SPC]) where *i,j,k* are indexes referencing to three dimensions: *i* varies be-

tween [1,5] corresponding to 5 horizontal values, j varies between [1,7] corresponding to 7 vertical values and k varies in [1,3] corresponding to 3 distance values. Equation 5.10 formulates the effect of the SPC parameter.

$$
\begin{aligned}
&position(X[i], Y[j], Z[k]) = position(X[i + SPC], Y[j + SPC], Z[k + SPC]) \\
&where, \\
&X = [XOppC, XCC, XC, XP, XEP] \\
&Y = [YLowerEP, YLowerP, YLowerC, YCC, YUpperC, YUpperP, YUpperEP] \\
&Z = [ZNear, ZMiddle, ZFar] \\
&i \in [1, 5] \quad j \in [1, 7] \quad k \in [1, 3] \\
&SPC \in [-1, 0, 1]
\end{aligned}
\tag{5.10}
$$

For example, given the robot wrist's position (X[3],Y[4], Z[1]) corresponding to (XC, YCC, ZNear): 1) if the SPC value is set to 1, the position becomes (X[4],Y[5], Z[2]) corresponding to (XP, YUpperC, ZMiddle); 2) if the SPC value is set to 0, there is no change; 3) if the SPC value is set to -1, the position becomes (X[2],Y[3], Z[1]) corresponding to (XCC, YLowerC, ZNear).

However, in order to maintain the meaning of the gesture, one or all of its dimensions may be fixed. For instance for a "stop" gesture as described in Listing 5.6, the distance and the vertical dimensions are resizable while the horizontal dimensions cannot be changed. This information is indicated when gestures are elaborated in the repository. In such a case the SPC parameter is applied only to the available gesture dimensions.

Listing 5.6: Certain fixed dimensions for gesture

```
<gesture lexeme="stop" mode="RIGHT_HAND">
    <phase type="STROKE">
        <hand distanceFixed="false"
              horizontalFixed="true"
              verticalFixed="false">
            <verticalLocation>YUpperC</verticalLocation>
            <horizontalLocation>XC</horizontalLocation>
            <locationDistance>ZMiddle</locationDistance>
            <handShape>form_open</handShape>
            <palmOrientation>AWAY</palmOrientation>
            <fingersOrientation>UP</fingersOrientation>
```

```
        </hand>
      </phase>
  </gesture>
```

Stroke Repetition - REP

The REP parameter defines the number of stroke phases in a gesture action. The duration of the whole gesture increases linearly with the REP value.

The REP parameter has a value between [-1, 1]. Depending on this value and the available time, the system decides how many stroke repetitions should be realized. Thus, this parameter modulates the duration of the stroke phase as well as the repetition number of stroke positions as described in Figure 5.24.

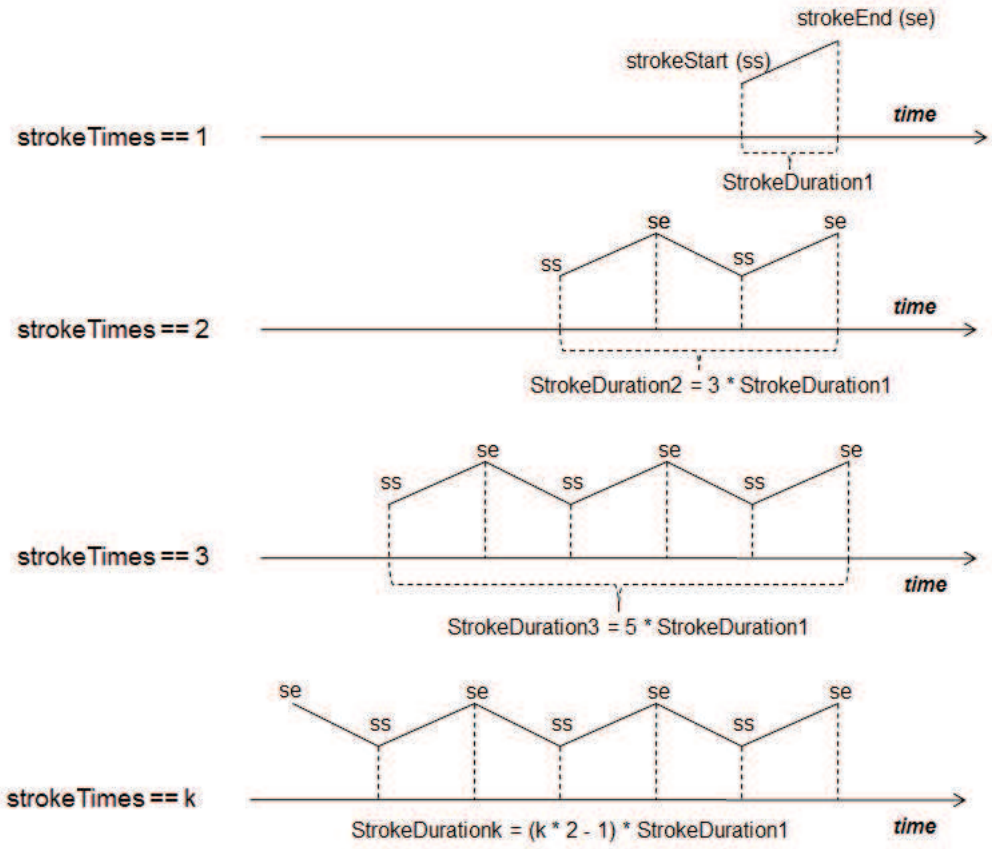


Figure 5.24: Stroke phase duration is modulated with the number of strokes

The duration of the stroke phase increases following Equation 5.11.

$$strokeDuration = (strokeTimes * 2 - 1) * strokeDuration \quad (5.11)$$

Summary

Table 5.7 shows a summary on the expressivity parameters which have been implemented for the Nao robot and the Greta agent so far.

EXP	Definition	Nao	Greta
TMP	Velocity of movement	Change coefficient of FittsŠ law	Change coefficient of FittsŠ law
SPC	Amplitude of movement	Limited in predefined key positions	Change gesture space scales
PWR	Acceleration of movement	No	Modulate stroke acceleration
REP	Number of stroke repetition times	Yes	Yes
FLD	Smoothness and Continuity	No	No
OPN	Relative spatial extent to body	No	elbow swivel angle
TEN	Muscular tension	No	No

Table 5.7: Implemented expressivity parameters

5.2.4 Keyframes Generation

This module calculates phases for the current gesture and creates corresponding gesture keyframes. Each keyframe contains the symbolic description, the timing and the expressivity parameters' values of one gesture phase. We use the same XML schema as used for the gesture description in the previous section. The symbolic representation allows us to use the same algorithms in the Behavior Realizer module for different embodiments.

In section 5.2.1 phases of a gesture are scheduled by giving real values to its sync points. The system creates one keyframe for each sync point as illustrated in Figure 5.25.

The surface form description of stroke's keyframes (i.e., strokeStart, strokeEnd) are copied from the gesture template taken out from the agent's gestuary. The description of keyframes is the same for Start and End sync points: that is a relaxation position of hand-arm gestures which is predefined ahead of time.

In the case of co-articulation for two gestures, the current gesture does not include the retraction phase. Consequently, there are no keyframes for the current gesture's end sync point and the next gesture's Start sync point.

keyframe[1]<time="Start", description of Start's position and motion, expressivity parameters>

keyframe[2]<time="Ready", description of Ready's position and motion, expressivity parameters>

keyframe[3]<time="StrokeStart", description of StrokeStart's position and motion, expressivity parameters>

keyframe[4]<time="StrokeEnd", description of StrokeEnd's position and motion, expressivity parameters>

keyframe[5]<time="Relax", description of Relax's position and motion, expressivity parameters>

keyframe[6]<time="End ", description of End's position and motion, expressivity parameters>

Figure 5.25: Gesture keyframes

In the case that there is a stroke repetition (i.e., the number of stroke times is more than 1), the number of keyframes increases: one more keyframe for each stroke sync point (i.e., strokeStart or strokeEnd) described in Figure 5.24.

5.3 Animation Generator for Nao

In the scope of this thesis work, we developed an animation generation module for the Nao robot as illustrated in Figure 5.26. Its objective is to generate joint values as animation parameters to be played by the robot. This module receives symbolic time-stamped keyframes as input generated by the Behavior Realizer module via the ActiveMQ messaging central system. We group keyframes per modalities (i.e., torso, head, gestures, speech) in order to create full body movements. In the case of gestures, each gesture keyframe includes a description for gesture phase, expressivity parameters, gesture trajectory type.

5.3.1 The Nao robot

First of all we need to show that, from a technical point of view, the Nao humanoid robot is capable of realizing expressive movements as well as producing multimodal signals (i.e., head, torso, speech, hand gestures) at the same time.

Nao is a 57 cm high humanoid robot with 25 degrees of freedom (i.e., DOF) and dynamic ranges of joint rotation (Figure 5.28).

A text-to-speech module provided by Acapela (www.acapela-group.com) is integrated within the robot so that it can speak with a customized voice. The robot

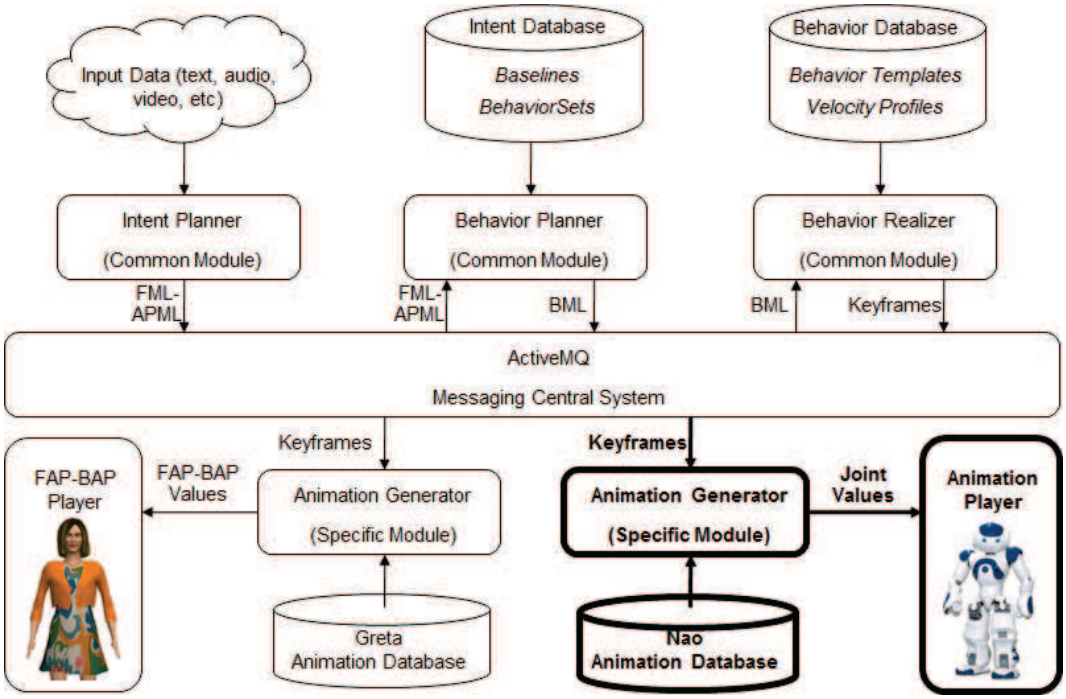


Figure 5.26: The Nao Animation Generator module is integrated within the GRETA system

can also control and play wave files through its loudspeakers.

Body	
Height (m)	0.57
Weight (kg)	4.5
Battery	
Type	Lithium-ion
Capacity	55 Wh
Degrees of freedom (DOF): 25	
Head	2 DOF
Arms	5 DOF X 2
Pelvis	1 DOF
Leg	5 DOF X 2
Hands	1 DOF X 2

Masses [g]	
Chest	1217.1
Head	401
Upper Arm	163
Lower Arm	87
Thigh	533
Tibia	423
Foot	158
Total	4346.1

Part	Motion	Range (°)
Leg (left)	hip twist (45°)	-68 to 44
	hip roll	-25 to 45
	hip pitch	-100 to 25
	knee pitch	0 to 130
	ankle pitch	-75 to 45
	ankle roll	-45 to 45
Arm (left)	shoulder roll	0 to 95
	shoulder pitch	-120 to 120
	elbow roll	-120 to 120
	elbow yaw	0 to 90
Head	yaw	-90 to 120
	pitch	-37 to 31

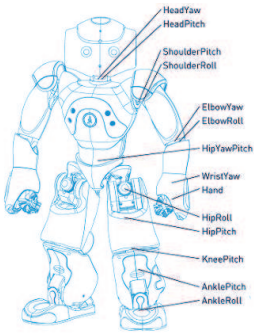


Figure 5.27: An overview of the Nao robot's specifications (Gouaillier et al., 2008)

Nao is a programmable robot that is supported with a graphical programming tool called Choregraphe and a set of available programming interfaces (i.e., SDK APIs) via a programming framework named NAOqi. It is designed to control the real robot with many high level programming languages such as C++, Python,

Matlab in multi-platform framework as Linux, Windows. The APIs can be executed through parallel, sequential or event-driven calls which allow the creation of complex plans to drive the robot (e.g., control different parts of body in parallel processes so that they can move at the same time).

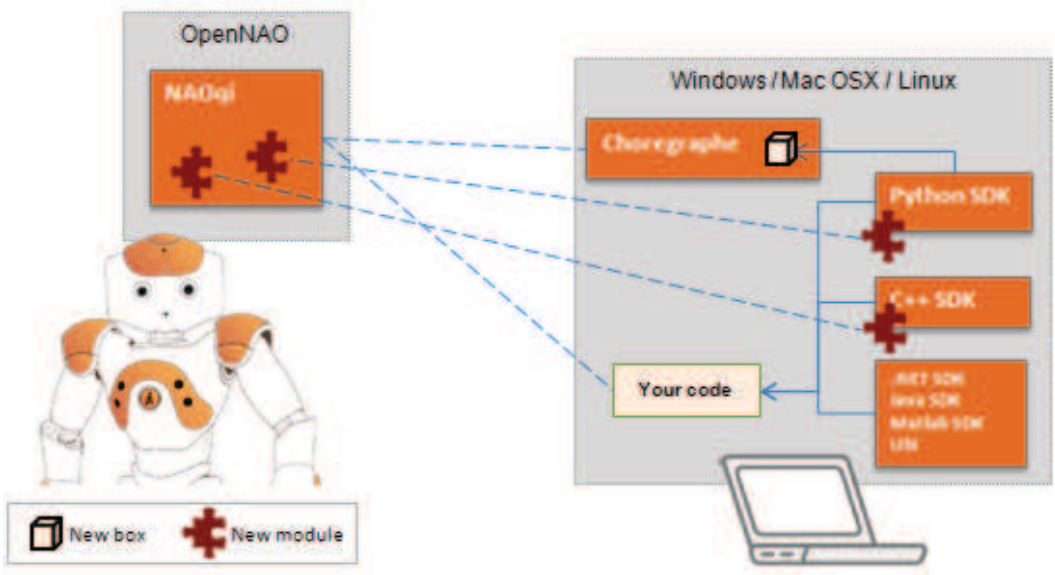


Figure 5.28: Tools for Nao programmers (Gouaillier et al., 2008; Pot et al., 2009)

Haring et al. (2011); Manohar et al. (2011) conducted an experiment on the Nao humanoid robot and showed that the robot has enough capacity to convey several emotions such as anger, happiness, sadness through its dynamic body movements. It means that it is possible for the robot to produce expressive gestures accompanying speech.

Limitations and solutions

The Nao robot has certain limitations which our controller module needs to overcome. Firstly, there are several singular positions in its gesture movement space that the robot hand can not reach. Secondly, its joints have a limited speed implying each movement has a minimum duration.

To deal with the first issue, we predefined a set of wrist positions that are feasible for the Nao robot. This task corresponds to building the Nao Gesture Space Specification (i.e., Animation Database) as illustrated in Figure 5.26. It

is based on a study of McNeill (1992) who claims that most gesture movements are realized in one gesture space. We followed his concentric gestural space as illustrated in Figure 2.4 to locate wrist positions for the robot. The coordinates in the Nao gesture space can take the following values:

- *Horizontal Location (X)* has 5 possible values (XEP, XP, XC, XCC, XOppC)
- *Vertical Location (Y)* has 7 possible values (YUpperEP, YUpperP, YUpperC, YCC, YLowerC, YLowerP, YLowerEP)
- *Frontal Location (Z)* has 3 possible values (ZNear, ZMiddle, ZFar)

The abbreviations stand for: C as Center, CC as Center-Center, P as Periphery, EP as Extreme Periphery, and Opp as Opposite Side corresponding to McNeill's space in Figure 2.4.

As a result, by combining these symbolical values (X, Z, Z) we have 105 positions in total to be defined in the Nao gesture space. Each symbolic position is instantiated with the corresponding wrist position in real values of 4 robot arm joints including *shoulder pitch*, *shoulder roll*, *elbow roll* and *elbow yaw* as shown in Table 5.8.

Code	X	Y	Z	Values (ShoulderPitch, ShoulderRoll, ElbowYaw, ElbowRoll)
keypos1	XEP	YUpperEP	ZNear	(-54.4953, 22.4979, -79.0171, -5.53477)
keypos2	XEP	YUpperEP	ZMiddle	(-65.5696, 22.0584, -78.7534, -8.52309)
keypos3	XEP	YUpperEP	ZFar	(-79.2807, 22.0584, -78.6655,-8.4352)
keypos4	XEP	YUpperP	ZNear	(-21.0964, 24.2557, -79.4565, -26.8046)
...	...	...	...	...

Table 5.8: Gesture Space Specification for the Nao robot

For the second issue (i.e., limited speed), we did an experiment to estimate the durations necessary to do hand movements in the Nao gesture space. This experiment was introduced in detail in the previous database section and resulted in Table 5.9.

5.3.2 Realtime Keyframes Synchronization

The system plans and realizes nonverbal multimodal behaviors in realtime. This means that keyframes of different modalities are generated and executed continuously. Our proposed algorithm controls keyframes of different modalities (e.g.,

Position (from-to)	keypos1	keypos2	keypos3	keypos4	...
keypos1	0.0	0.18388	0.28679	0.2270	...
keypos2	0.18388	0.0	0.19552	0.2754	...
keypos3	0.28679	0.19552	0.0	0.3501	...
keypos4	0.2270	0.2754	0.3501	0.0	...
...	...	...	...	...	...

Table 5.9: Minimum durations necessary to do a hand-arm movements between any two positions (coded as keypos1, keypos2, etc) in the Nao gesture space

torso, gesture, head, etc) in parallel processes. Hence verbal and nonverbal multimodal behaviors of the same communicative intent are realized at the same time. These processes are separated but synchronized with each other by receiving timing information from the central clock of the global system as illustrated in Figure 5.29.

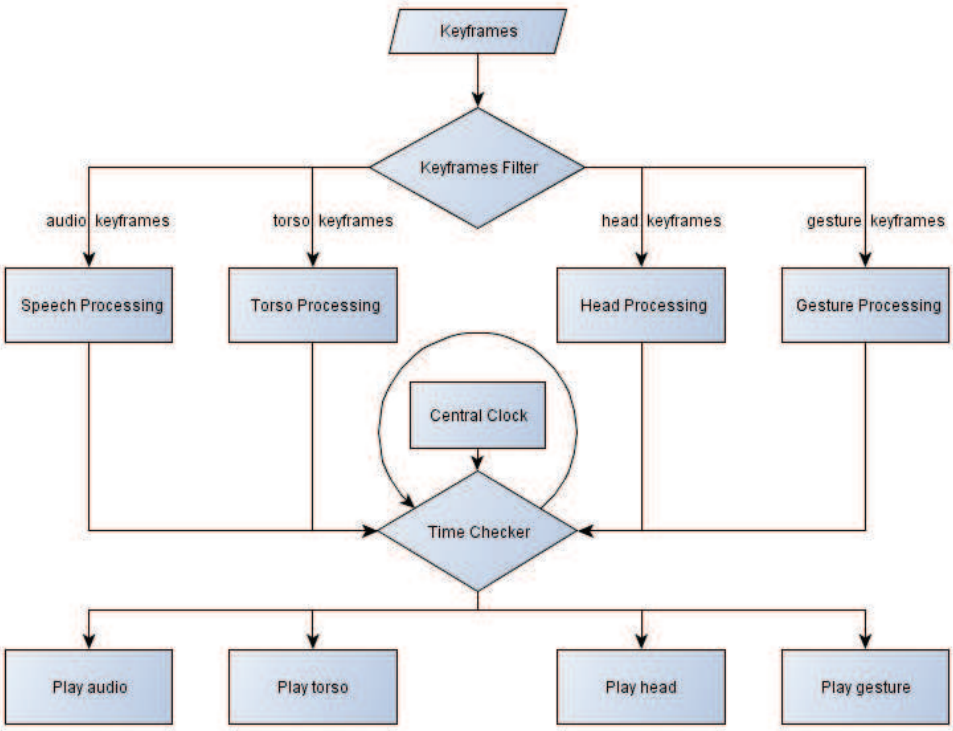


Figure 5.29: Synchronize different signals using parallel processes: one process is created for each modality (speech, torso, head, gesture). These processes synchronize each other by updating the timing information from the same central clock.

A module of Keyframes Filter is developed to receive and distribute coming keyframes into modality dependent processes correspondingly. Each process (i.e., Torso Procedure, Head Procedure, Gesture Procedure) computes the animation parameters from its keyframes. It then plays animations at the right time determined by the central clock of the global system via a Time Checker module. The speech process sends sound data to the robot's loudspeaker and plays it at the planned time.

5.3.3 Keyframe Gesture Processing

We want to compute robot hand movements from gesture keyframes. To do that, animation parameters have to be computed from a keyframes interpolation. An issue rises when the system interpolates and plays all keyframes at once: while the animations are being processed, the system cannot be interrupted. Hence, another solution is to divide the keyframes into smaller sequences and play each sequence of gestures one after the other. The problem is now: How to divide keyframes for the interpolation?

As mentioned in the gesture scheduling section, co-articulation happens between consecutive gestures. If a gesture is co-articulated with the next gesture, then its retraction phase is canceled. We define a *gesture trajectory* as a sequence of consecutive gestures which satisfy following conditions: 1) the first gesture starts from a rest position; and 2) there is only one retraction phase at the last gesture in the sequence. Such a gesture trajectory corresponds to what is called G-Unit of (Kendon, 1980). The description of a gesture trajectory is illustrated in Figure 5.30.

Keyframes belonging to the same gesture trajectory will be interpolated and played together. This solution ensures smoothness of movements within a gesture trajectory. It also allows the system to insert an interruption command after executing a gesture trajectory. The algorithm to do this solution is described in Figure 5.31.

In practice, the algorithm is implemented with the Nao robot built-in proprietary procedures (Gouaillier et al., 2008). One of them is the API *angleInterpolation* as described in Figure 5.32.

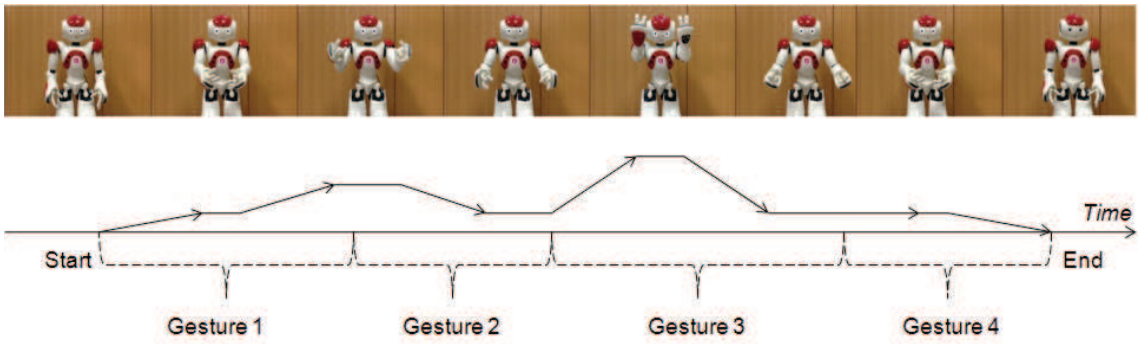


Figure 5.30: A Nao gesture trajectory of four consecutive gestures: the first gesture starts from a rest position and there is only one retraction phase at the last gesture

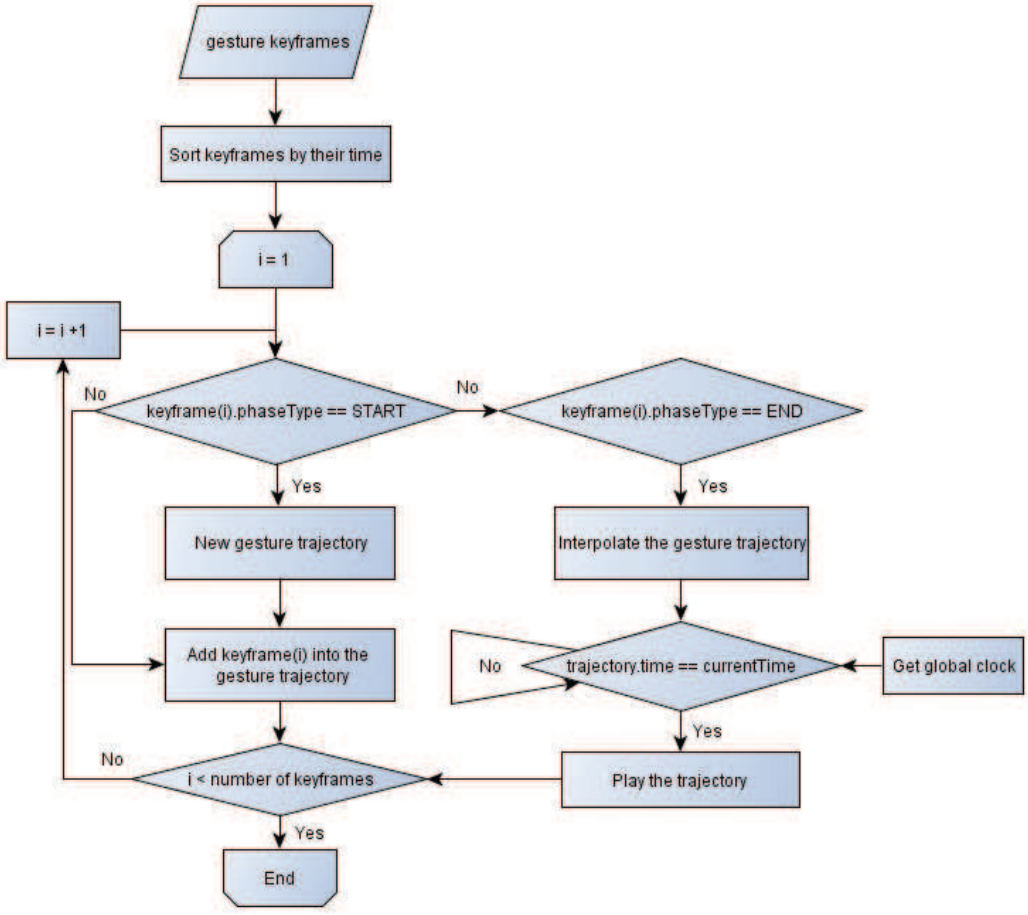


Figure 5.31: Algorithm to find and interpolate gesture trajectories

```
void ALMotionProxy::angleInterpolation(const AL::ALValue& names, const
AL::ALValue& angleLists, const AL::ALValue& timeLists, const bool&
isAbsolute)
```

Interpolates one or multiple joints to a target angle or along timed trajectories. This is a blocking call.

Parameters:
- *names* – Name or names of joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
- *angleLists* – An angle, list of angles or list of list of angles in radians
- *timeLists* – A time, list of times or list of list of times in seconds
- *isAbsolute* – If true, the movement is described in absolute angles, else the angles are relative to the current angle.

Figure 5.32: Description of the Nao SDK API *angleInterpolation*

5.4 Conclusion

We have presented the algorithms and technical methods used in our expressive gesture model. The modules implemented in this thesis work has been conducted in the framework of the GRETA multimodal behavior generation system. These modules include a Behavior Realizer and a Nao Animation Realizer. Additionally, a gesture representation language has been designed to specify gesture templates in a gestuary.

The Behavior Realizer module can be considered as a BML realizer as it deals with the standard version of the BML language in the SAIBA framework.

The model uses a formulation of the Fitts' law to simulate the speed of human gestures. The duration of gesture phases are computed using this formulation. From these calculated phase durations, the system schedules exactly the timing of gestures via its sync points accord linked to speech timing. In the gesture scheduling module, we have proposed concrete algorithms to handle the issue of co-articulation between gestures while maintaining the synchronization of gestures and speech.

The implementation of the gesture expressivity is important in our gesture model. Its expressivity parameters such as the temporal extent, the spatial extent

and the stroke repetition change the gesture timing as well as the gesture trajectory.

For the Nao humanoid robot module, in order to overcome its physical constraints, we have defined a gesture space specific to its gestures. This gesture space contains key positions the robot hands can reach while gesturing. Additionally, we have conducted an experiment to estimate minimum durations of robot hand movements for any two positions in the predefined gesture space. Then the model uses these results to generate gestures that are feasible for the robot.

The module to control Nao gestural animation has been designed and implemented in a parallel processing architecture. This allows the module to receive and realize each signal modality in a separated process. Thus, two modalities like gesture signal and speech signal can be played synchronously. The interpolation between keyframes sent to the robot is done by grouping keyframes of the same gesture trajectory. A gesture trajectory is defined as a sequence of gestures which are co-articulated with each other.

There are still several tasks to be completed in the implementation. Firstly a concrete value for the maximal speed limitation of human gestures has not yet been instantiated. Secondly, the difference in velocity between stroke phase and other phases need to be studied. Thirdly, the realization of certain expressivity parameters like Fluidity, Tension, Power is still a challenge left for future work.

Chapter 6

Evaluation

To validate the developed expressive gesture model, we conducted a perceptive experiment. This chapter presents our design and implementation of this experiment as well as its obtained results.

We wanted to evaluate how gesture expressivity and gesture timing of a robot were perceived by human users. Through this experiment, we evaluated the quality of gesture expressivity for the dimensions of spatial extent and temporal extent as well as the quality of the temporal coordination between gesture and speech. Additionally, we wanted to study some effects of robot gestures in communication such as whether the robot can convey more information through its expressive gestures (e.g., information complementary or redundant to the one indicated by speech).

The results of this experiment could be used not only to validate our expressive gesture model for a humanoid robot, but also to answer the research question: "Whether a physical robot can display gestures with expressivity?".

The implementation of our experiment followed several steps such as defining a protocol to conduct the experiment, building materials and proposing hypotheses, defining procedures and collecting participants, analyzing and discussing obtained results as described in Figure 6.1. The detail of this implementation will be presented in the following sections.

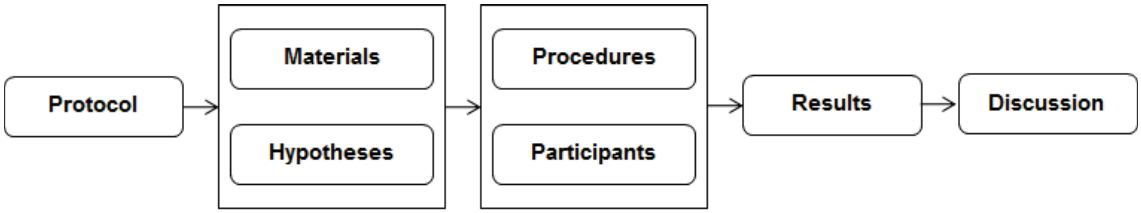


Figure 6.1: Steps to conduct our perceptive evaluation for our expressive gesture model

6.1 Protocol

We designed and implemented a perceptive experiment in which participants rated gestures displayed by the Nao robot (Gouaillier et al., 2009) while it was telling excerpts of the French tale: "Three pieces of night".

To do that, a gestuary was built in such a way that it was feasible for the robot. This gestuary contained gestures that were encoded using our defined gesture specification. The elaboration of these gestures was based on gesture annotations extracted from a storytelling video corpus of real actors who were telling the same story: "Three pieces of night" (Martin, 2009). Our system selected gestures from this gestuary and generated robot expressive gestures in different conditions:

Condition C1 Gestures are synchronized with speech: Robot gestures were scheduled using our gesture scheduling module integrated within the GRETA system.

Condition C2 Gestures are asynchronized with speech: Half of given gestures were set to be desynchronized with speech. This task was done by modifying the gesture timing manually to arrive late by 0.5 s. We did not setup the asynchronization for all gestures to avoid the lost the robot's credibility after a few utterances.

Condition C3 Gestures are synchronized and produced with expressivity: The values of expressivity parameters are set to be in accord with emotional states of the characters in the story which is being told by the robot. These emotional states were annotated manually ahead of time (Doukhan et al., 2011).

Condition C4 Gestures are synchronized and produced without expressivity: The values of expressivity parameters are set to zero (i.e., neutral state).

The story "Three Pieces of Night" was divided into 8 segments. Four of them containing emotional elements in the story were used to test the gesture expressivity. We created 4 pairs of videos of robot gestures in conditions (C3, C4). Similarly, we created 4 pairs of videos for the 4 other segments of the story to test the synchronization of gestures with speech in conditions (C1,C2). The last video was created while the robot was telling the whole story in a synchronized and expressive manner (i.e., condition C3). This video was used to test general effects of the robot gestures in communication. The same position of camera, lighting and quality of video encoding were used in all videos.

Consequently, we have three group tests of trials. The first test (T1) included pairs of videos in conditions C1 and C2 to test the synchronization of gestures with speech. The second test (T2) included pairs of videos in conditions C3 and C4 to test the gesture expressivity. The third test (T3) included only one long video in condition C3.

The participants watched pre-recorded videos of the robot through a graphical interface and then answered a multiple choice questionnaire below each video. The experiment took place online through Internet. This method allowed us to collect a large number of participants with a low cost. We used the open source LimeSurvey developed by Schmitz (2010) to create user interfaces and to save data.

The results obtained from the experiment was analyzed using the one-way ANOVA test to compare means of two gesture groups in conditions (C1 vs. C2) or (C3 vs. C4).

6.2 Materials

The text of the story was analyzed and annotated by Doukhan et al. (2011) and then encoded within a FML message as illustrated in Listing 6.1. Our system took as input this FML message and produced expressive gestures and voice for the Nao robot. The expressive voice of the robot in French was generated using the text-to-speech synthesizer Alcapela (www.acapela-group.com) which is integrated in our system.

Listing 6.1: An example of FML-APML generated by Doukhan et al. (2011)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <!DOCTYPE fml-apml SYSTEM "fml-apml.dtd" []>
    <fml-apml>
        <bml>
                <speech id="s1" start="0.0">
                    <text>
                        ....
                        <sync id='tm49'/>
                        \vce=speaker=Antoine\ \rspd=100\ \vol=26214\ \vct=100\
                        se dit dame Souris,
                        \pau=500\
                        \vce=speaker=AntoineSad\ \rspd=80\ \vol=32767\
                        <sync id="tm50"/>
                        La vie est bien compliquée.
                        <sync id="tm51"/>
                        \vce=speaker=Antoine\
                        \pau=350\
                        \vce=speaker=AntoineHappy\ \rspd=110\ \vol=32767\
                        Mais il y a
                        <sync id="tm52"/>
                        toujours moyen de s'arranger.
                        ....
                    </text>
                </speech>
        </bml>
        <fml>
            ...
            <performative type="complain" character="sad_profile" id="p22" start
                ="s1:tm50" end="s1:tm51"/>
              <emotion type="happy" character="happy_profile" id="p23" start="s1:
                  tm51" end="s1:tm52"/>
            ...
        </fml>
    </fml-apml>
```

To express different emotional states of characters in the story through gesture performance, we elaborated four gesture profiles (i.e., four sets of the expressivity parameters) corresponding to four voice variants of Alcapela (i.e., sad voice,

happy voice, angry voice and neutral voice). The values attributed to expressivity parameters in these sets were based on studies on the relation between emotional states and quality of body movement as shown in Table 6.1. As a result, we have a table of expressivity parameters' values as showed in Table 6.2.

State	Description
Happy/ Joy	Movements in happy displays were judged to be expanded, and to be action-filled, loose, fast, relatively soft and somewhat jerky, also consistent with previous research (Aronoff et al., 1992; Boone and Cunningham, 1996; Meijer, 1989; Montepare et al., 1999; Wallbott et al., 1986; Argyle, 1988)
Sadness/ Boredom/ Tired	Movements in displays of sadness lacked action and were rated as relatively contracted, soft, and smooth, as reported in previous studies (Meijer, 1989; Montepare et al., 1999; Wallbott et al., 1986; Argyle, 1988)
Angry	Movements in angry displayed were characterized by variations in velocity and force, and accompanied by abrupt changes in tempo and direction, as well as angularity or sharpness in body form (i.e., very jerky, stiff, hard, fast, and action-filled), as suggested by previous research (Aronoff et al., 1992; Boone and Cunningham, 1996; Meijer, 1989; Montepare et al., 1999; Wallbott et al., 1986; Argyle, 1988; Gallaher, 1992)

Table 6.1: The way of gesture movements (i.e., gesture expressivity) is linked to emotional states

Emotion	SPC	TMP
Neutral	Medium	Medium
Happy	High	High
Sad	Low	Low
Angry	Medium	High

Table 6.2: Expressivity parameters are linked to emotional states from Table 6.1: Medium=0, High=1, Low=-1

The system selected and planned on the fly expressive gestures from a set of symbolical gesture templates. The elaboration of these gestural templates stored in the Nao gestuary was based on gesture annotations extracted from the Story-telling Video Corpus. Such gesture templates were built following three steps as illustrated in Figure 6.2.

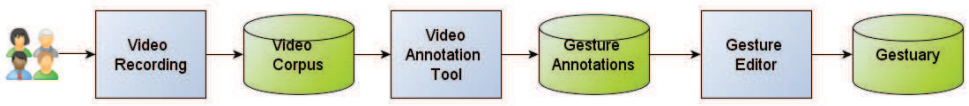


Figure 6.2: Three steps to elaborate gesture prototypes

Step 1: The video corpus had been constructed by Martin (2009), a partner of the GVLEX project. To do this corpus, six actors were videotaped while telling

twice the French story: "Three Pieces of Night". Two cameras were used (front and side views) to get postural expressions in the three dimensional space.

Step 2: Gestures were annotated with the Anvil video annotation tool (Kipp et al., 2008) as illustrated in Figure 6.3. This task was also done by Martin (2009). Each gesture of the actors was annotated with information of its category (i.e., iconic, beat, metaphoric and deictic), its duration and which hand were used.

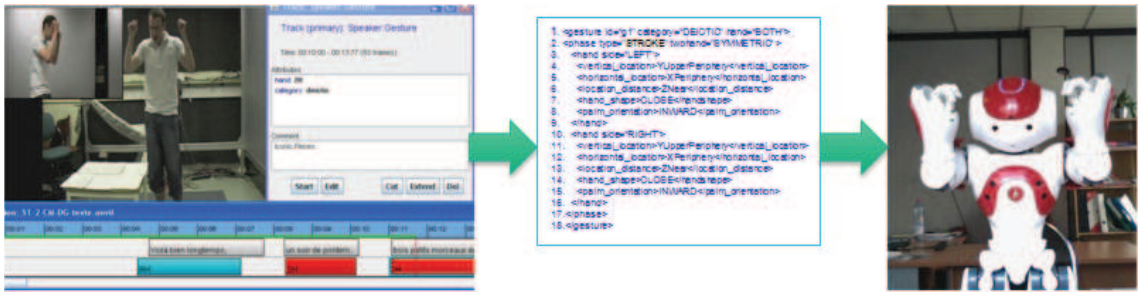


Figure 6.3: From Anvil gesture annotations to robot gestures

Step 3: From the form of gestures displayed on the videos with their annotated information, we have elaborated symbolic gesture templates respectively. These gestures were verified ahead of time to ensure that they were feasible by the Nao robot (see Figure 6.4).

6.3 Hypotheses

We hypothesized that robot gestures controlled by our system would satisfy at least two critical points: (1) They are tightly tied to speech; (2) They are expressive. Two hypotheses to be tested as following:

H1 Participants rate the temporal coordination between robot gestures and speech more positively when the gestures are produced in condition C1 than when the gestures are produced in condition C2.

H2 Participants rate the expressivity of robot gestures more positively when the gestures are produced in condition C3 than when the gestures are produced in condition C4.

Figure 6.4: Some Nao's gestures are generated for the story "Three pieces of night"

In addition, four supplementary hypotheses are considered related to the findings that gestures and speech communicate complementary, redundant or event contradictory information (McNeill, 1985; Kendon, 2004; Bergmann et al., 2011).

H3 Participants' perceptions of naturalness of robot gestures will be higher when

the gestures are synchronized with speech than when the gestures are not synchronized with speech.

H4 Participants have a better impression that relevant gestures convey complementary information to the one indicated by speech when the gestures are produced with expressivity than when the gestures are produced without expressivity.

H5 Participants have a better impression that relevant gestures convey contradictory information to the one indicated by speech when the gestures are produced with expressivity than when the gestures are produced without expressivity.

H6 Participants have a better impression that relevant gestures convey redundant information to the one indicated by speech better when the gestures are produced with expressivity than when the gestures are produced without expressivity.

6.4 Procedure

Participants were invited to evaluate the Nao storyteller robot through an invitation email. If they accepted to do the evaluation, they opened the evaluation's homepage with their browser (i.e., Firefox, IE, Chrome, etc) through an available Internet link in the email. The evaluation was done with videos of the robot.

The evaluation's homepage was a short introduction in French to the participants such as our objective and the estimated time to do the whole evaluation as shown in Figure 6.5.

Then they were asked to enter certain personal information such as age, sex, education level, culture, profession (Figure 6.6).

The experiment started with a trial that was given through an graphical interface containing a video and a multiple choice questionnaire as illustrated in Figure 6.7.

The participants clicked the play button to watch the video. They could replay the video as many times as they wanted. Then, they answered the questionnaire

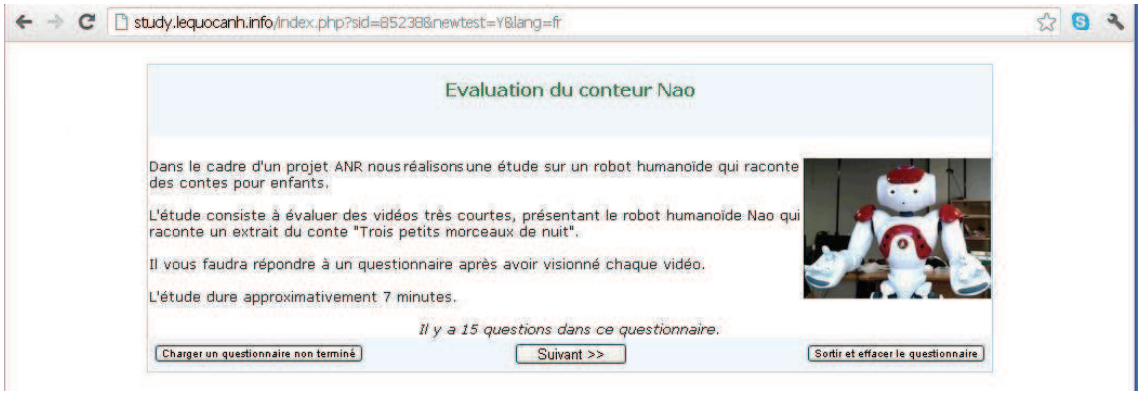


Figure 6.5: The Internet interface to introduce evaluation

below the video. Regarding the questionnaire, we asked the participants how much they agreed or disagreed with statements related to robot gestures on a seven-point likert scale from strongly disagree to strongly agree as illustrated in Figure 6.8.

The participants were obligated to finish all the questions in a trial to enable the *next* button. Once they clicked the next button to do an other trial, they could not go back to the previous trial.

There were totally 9 trials shown to the participants. The first four trials were used to evaluate the gesture timing (i.e., test group T1). The next four trials were used to evaluate the gesture expressivity (i.e., test group T2). In both set of trials, the order of presentation of the 4 videos was selected randomly to avoid any effect of one video on other ones. And the last trial was used to evaluate general effects of robot gestures in a long time (i.e., test group T3).

In the test group T1, each trial displayed one short video(i.e., an excerpt of the selected story) that was produced either in condition C1 (i.e., gestures are synchronized with speech) or in condition C2 (i.e., gestures are asynchronized with speech). The system selected one from two videos in a pair (C1, C2) randomly. The participants evaluated the video without knowing in what condition the video was produced. Two statements to be rated by the participants using the seven-point likert scale in each trial of T1 were:

- *Gesture were synchronized with speech*
- *Gestures seemed natural*

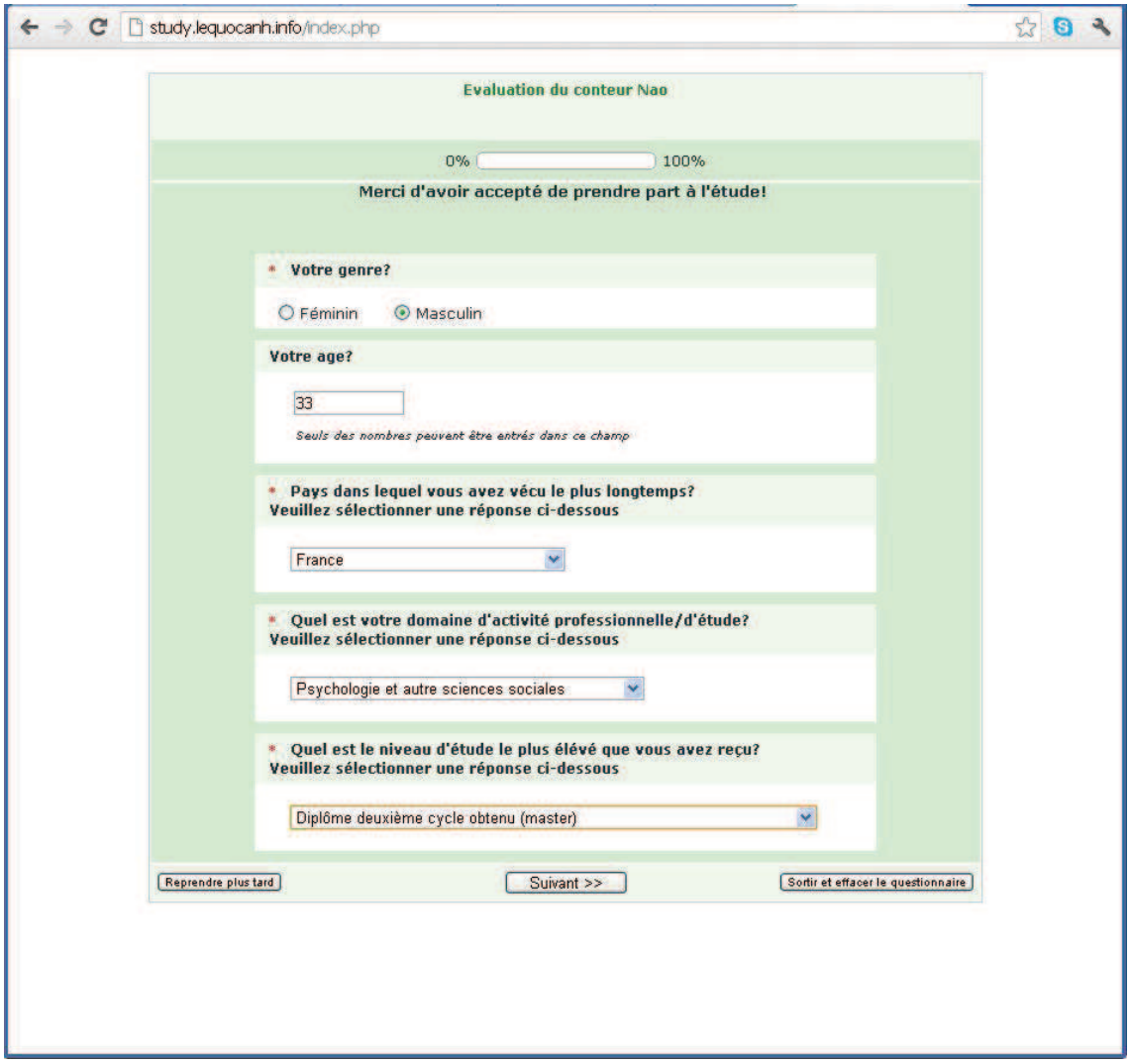


Figure 6.6: The evaluation interface to collect participants' personal information

Similarly to the group T1, each trial in the group T2 displayed one short video that was produced either in condition C3 (i.e., gestures are produced with expressivity) or in condition C4 (i.e., gestures are produced without expressivity). The graphical interface of the trials in the group T2 is illustrated in Figure 6.9. Four statements to be rated by the participants using the seven-point likert scale in each trial of T2 were:

- *Gestures were executed with expressivity*

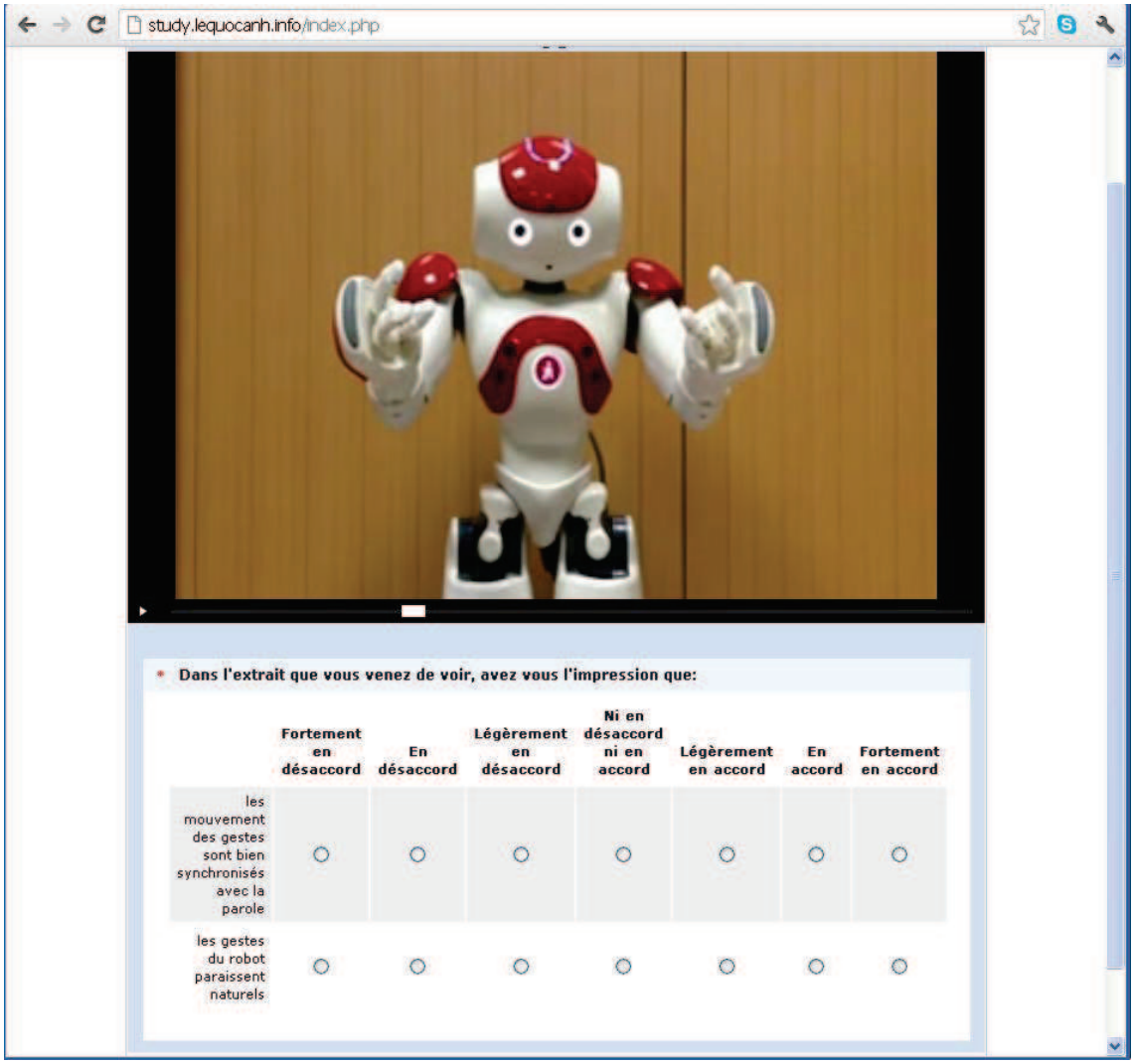


Figure 6.7: An interface of the test group T1

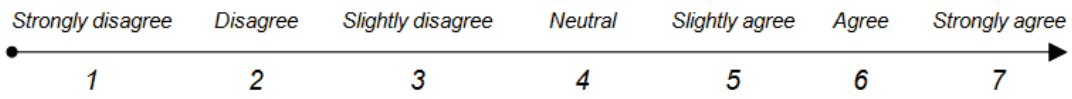


Figure 6.8: The seven-point likert scale was used to rate participants' perceptions

- *Gestures and speech provided contradictory information*
- *Gestures and speech provided complementary information*
- *Gestures and speech provided redundant information*

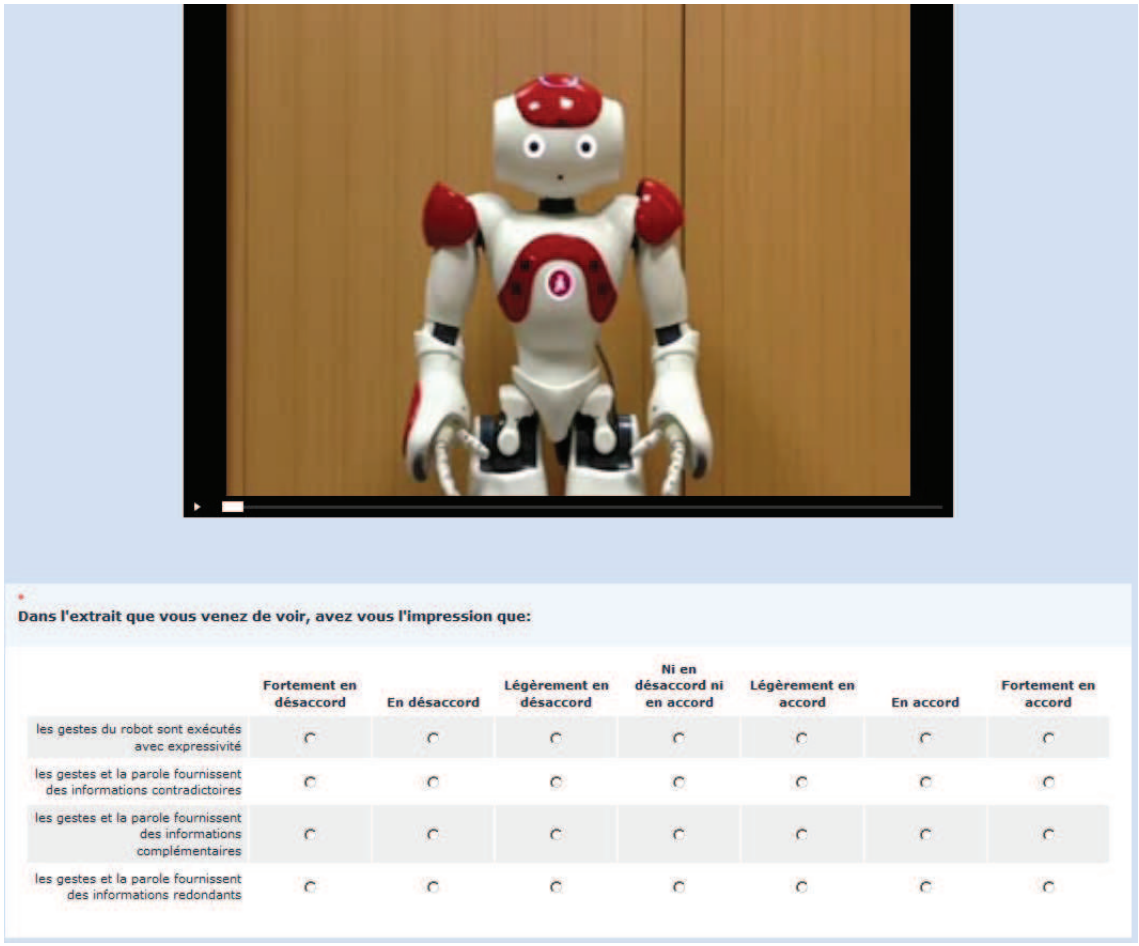


Figure 6.9: An interface of the test group T2

The last trial in the group T3 displayed a long video which lasted 2.51 minutes in which the robot produced gestures in condition C1 and C3 (i.e., expressive gestures are synchronized with speech) while telling the whole story "Three pieces of night". Five statements to be rated by the participants using the seven-point likert scale in this trial were:

- *Gestures were synchronized with speech*
- *Gestures were executed with expressivity*
- *Gestures and speech provided contradictory information*
- *Gestures and speech provided complementary information*

- *Gestures and speech provided redundant information*

In addition, in order to test how the story spoken by the robot is attractive to the participants, we asked them whether they wanted to see another video of the robot telling another tale. The graphical interface of the trial in the group T3 is illustrated in Figure 6.10.

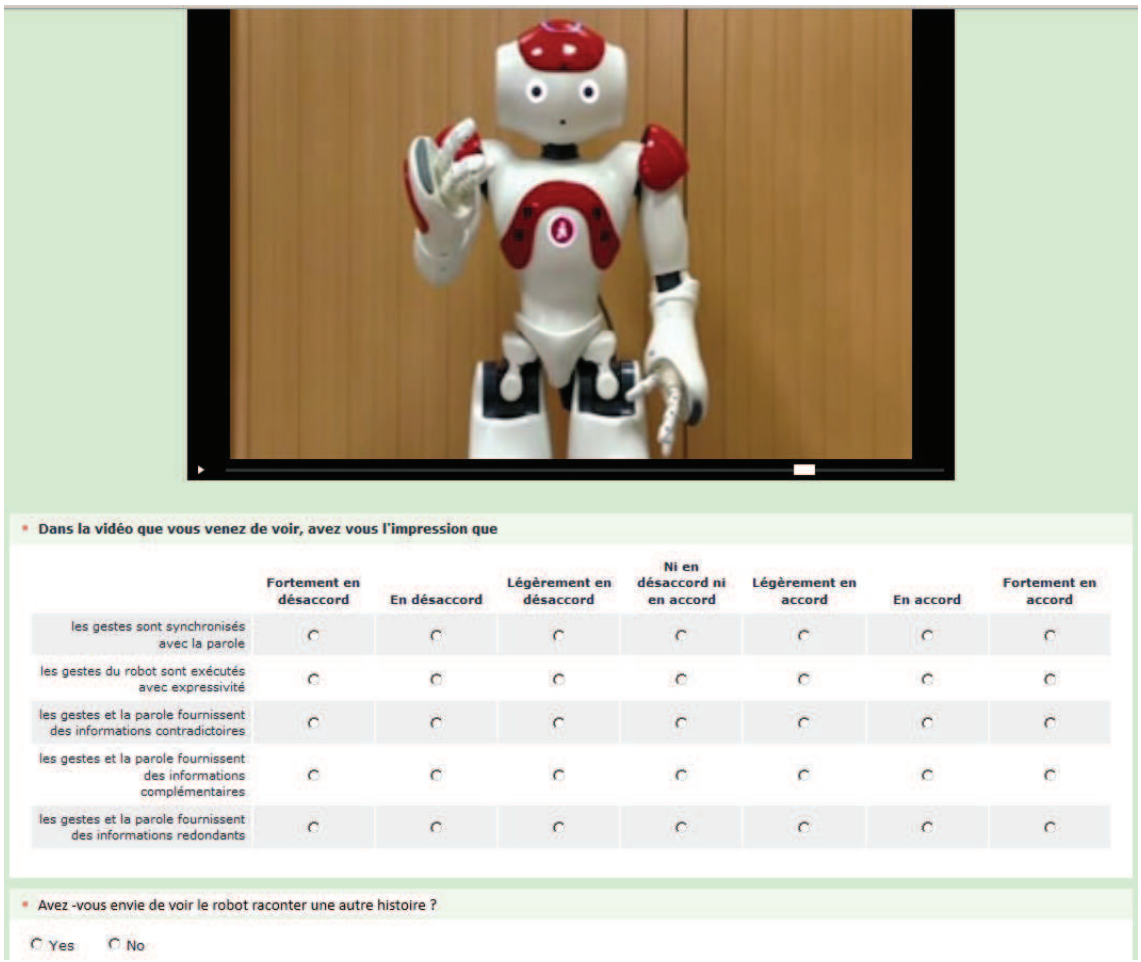


Figure 6.10: The interface of the test T3

As the robot spokes French and our experiment was dedicated to French speakers, all the questions were written in French. The average duration to do the evaluation was about 7 minutes but there was no time limit for participants.

6.5 Participants

Sixty three participants (27 females and 36 males) took part and answered completely the questionnaires in our experiment. The ages of the participants varied between 23 and 67 years (Mean = 37.02, Standard Deviation = 12.14). All participants were French speakers collected from the university of Telecom ParisTech through our invitation emails. Participants were mainly students and staffs at Master (15 participants), PhD (37 participants) and other (11 participants) levels. They were working and studying in different domains of the university including computer science, electronics, economic and social sciences as well as psychology.

6.6 Results

This section is dedicated to describe our analysis from the subjective data obtained through our experiment. The interpretation and discussion of the results will be left to the next section.

We used Analysis of Variance (ANOVA) tests to analyze the data. Group mean values were calculated and a between-subject ANOVA was carried out using a decision criterion of 0.05 (i.e., there will be a significant effect of obtained results if the probability value in ANOVA test, p-value < 0.05).

The following subsections presents the results in detail.

Results for the hypothesis H1: Gestures and speech synchronization

Table 6.3 reports the results obtained from the test group T1. It shows that the participants rated the robot gestures generated in condition C1 (M = 3.60, SD = 2.01) higher than the robot gestures generated in condition C2 (M = 3.07, SD = 1.48). Analysis of variance (ANOVA) comparing the two experimental groups (synchronization vs. asynchronization) with regard to ratings of gesture timing shows a significant effect (F = 4.94, $p < 0.05$).

Criteria to be rated	Number of participants	Mean of ratings (M)	Standard Deviation (SD)
Gesture timing in C1	63	3,60	2,01
Gesture timing in C2	63	3,07	1,48

Table 6.3: Perceptive results on gesture timing

Results for H2: The expressivity of gestures

Table 6.4 reports the results obtained from the test group T2 for the hypothesis H2. It shows that the participants rated the robot gestures generated in condition C3 (M = 4.80, SD = 1.20) higher than the robot gestures generated in condition C4 (M = 4.38, SD = 1.31). The analysis of variance (ANOVA) comparing the two experimental groups (gestures with and without expressivity) with regard to ratings of gesture expressivity shows a significant effect ($F = 4.42$, $p < 0.05$).

Criteria to be rated	Number of participants	Mean of ratings	Standard Deviation
Gesture expressivity in C3	63	4,80	1,20
Gesture expressivity in C4	63	4,38	1,31

Table 6.4: Summary for perceptive results on gesture expressivity

Results for H3: The naturalness of gestures

Table 6.5 reports the results obtained from the test group T2 for the hypothesis H3. It shows that the participants rated the naturalness of the robot gestures generated in condition C1 (M = 3.03, SD = 2.25) slightly higher than the naturalness of the robot gestures generated in condition C2 (M = 2.93, SD = 1.93). However, the analysis of variance (ANOVA) comparing the two experimental groups (gestures with and without expressivity) with regard to ratings of gestural naturalness does not show a significant effect (F = 0.1, N.S.).

Criteria to be rated	Number of participants	Mean of ratings	Standard Deviation
Naturalness of gestures in C1	63	3,03	2,25
Naturalness of gestures in C2	63	2,93	1,93

Table 6.5: Summary for the perceptive results for gesture naturalness

Results for H4: Gestures and speech convey complementary information

Table 6.6 reports the results obtained from the test group T2 for the hypothesis H4. It shows that the participants rated the naturalness of the robot gestures generated in condition C3 (M = 4.07, SD = 1.17) slightly higher than the naturalness of the robot gestures generated in condition C4 (M = 4.10, SD = 1.00). However, the analysis of variance (ANOVA) comparing the two experimental groups

(gestures with and without expressivity) with regard to ratings of gesture-speech complementarity does not show a significant effect (F = 0.02; N.S.).

Criteria to be rated	Number of participants	Means of ratings	Standard Deviation
Gesture-speech complementarity in C3	63	4,07	1,17
Gesture-speech complementarity in C4	63	4,10	1,00

Table 6.6: Summary for the perceptive results for H4

Results for H5: Gestures and speech convey contradictory information

Table 6.7 reports the results obtained from the test group T2 for the hypothesis H5. It shows that the participants rated the naturalness of the robot gestures generated in condition C3 (M = 3.24, SD = 1.16) slightly higher than the naturalness of the robot gestures generated in condition C4 (M = 3.15, SD = 0.99). However, the analysis of variance (ANOVA) comparing the two experimental groups (gestures with and without expressivity) with regard to ratings of gesture-speech contradictory does not show a significant effect (F = 0.22; N.S.).

Criteria to be rated	Number of participants	Mean of ratings	Standard Deviation
Gesture-speech contradictory in C3	63	3,24	1,16
Gesture-speech contradictory in C4	63	3,15	0,99

Table 6.7: Summary for the perceptive results for H5

Results for H6: Gestures and speech convey redundant information

Table 6.8 reports the results obtained from the test group T2 for the hypothesis H6. It shows that the participants rated the naturalness of the robot gestures generated in condition C3 (M = 3.71, SD = 1.36) slightly higher than the naturalness of the robot gestures generated in condition C4 (M = 3.50, SD = 1.08). However, the analysis of variance (ANOVA) comparing the two experimental groups (gestures with and without expressivity) with regard to ratings of gesture-speech redundant does not show a significant effect (F = 1.17; N.S.).

Results for the whole story

Regarding the perceptions of the participants for the whole story, the robot gestures were evaluated as acceptable for the synchronization of gestures with speech

Criteria to be rated	Number of participants	Mean of ratings	Standard Deviation
Gesture-speech redundant in C3	63	3,714285714	1,368663594
Gesture-speech redundant in C4	63	3,5	1,088709677

Table 6.8: Summary for the perceptive results for H6

(M = 4.97, SD = 1.04) and for the gesture expressivity (M = 4.78, SD = 1.21) (see Figure 6.11).

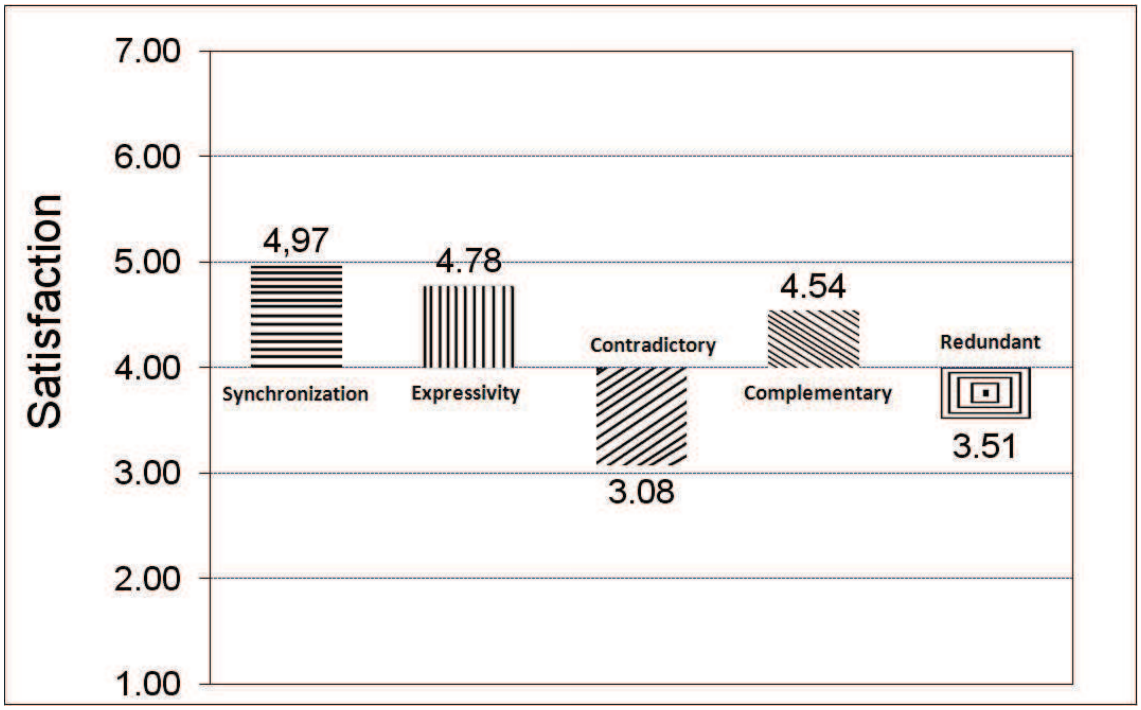


Figure 6.11: The results from our analysis of the subjective data from the experiment for whole story: "Three small pieces of night" using the likert scale: (1 = Strongly disagree, 2 = Disagree, 3 = Slightly disagree, 4 = Neutral, 5 = Slightly agree, 6 = Agree, 7 = Strongly agree)

There were 48 participants (76%) who agreed that the robot gestures were synchronized well with speech, in which 23 participants (36%) gave a slight agreement and 25 participants (40%) gave an agreement or a strong agreement. Regarding gesture expressivity, 44 participants (70%) agreed that the robot gestures were expressive of which 24 participants (38%) gave a slight agreement and 20 participants (32%) gave an agreement or strong agreement. These statistic results are

illustrated in Figure 6.12.

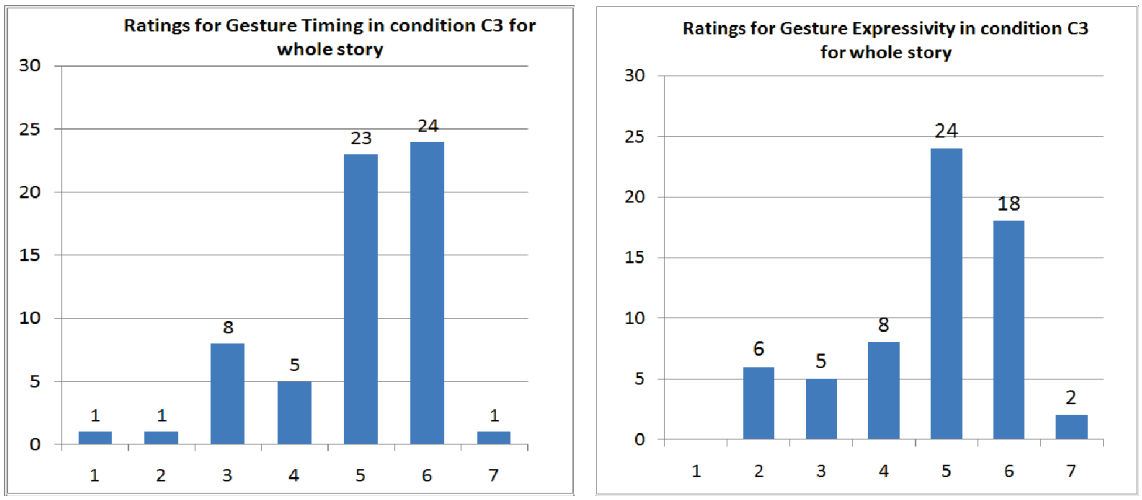


Figure 6.12: The participants' rates for gesture timing using the likert scale: (1 = Strongly disagree, 2 = Disagree, 3 = Slightly disagree, 4 = Neutral, 5 = Slightly agree, 6 = Agree, 7 = Strongly agree)

In addition, the data obtained from the perceptive tests for the hypotheses (H4, H5, H6) that gestures and speech communicate complementary, redundant or contradictory information were analyzed.

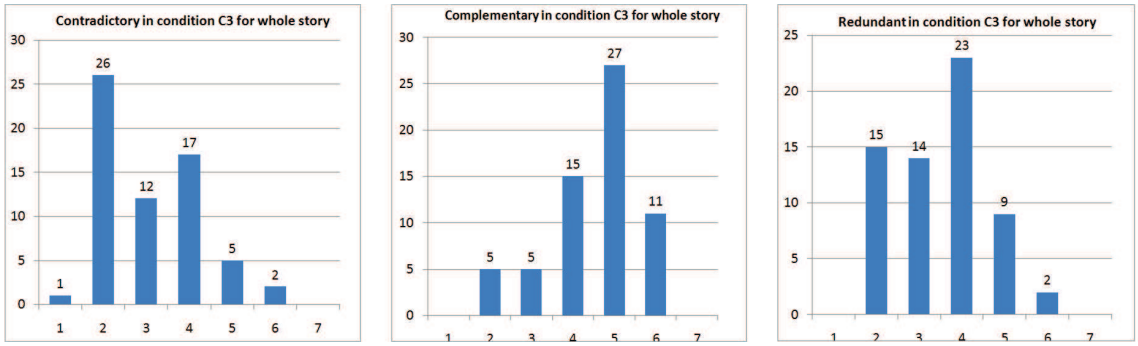


Figure 6.13: The results from our analysis of the subjective data from the experiment for whole story: "Three small pieces of night" using the likert scale: (1 = Strongly disagree, 2 = Disagree, 3 = Slightly disagree, 4 = Neutral, 5 = Slightly agree, 6 = Agree, 7 = Strongly agree)

Thirty eight participants (60%) agreed that the robot gestures were complementary to the information indicated in the speech content (M = 4.54, SD = 1.17).

Most of participants did not perceive that the gestures and speech conveyed contradictory (M = 3.08, SD = 1.30) nor redundant information (M = 3.51, SD = 1.12). Figure 6.13 reports the results.

Finally we analyzed data from the last question "Would you like to watch the robot tells another story?" to test whether the robot was attractive to the participants. The results showed that eighteen participants (29%) were ready to see more videos.

6.7 Interpretation and Discussion

The main goal of this experiment was to evaluate how the Nao robot gestures controlled by our expressive gesture model were perceived by human users.

The experiment was divided into two stages. In the first stage, the participants gave their opinion on the performances of gestures in the short videos (i.e., short excerpts of the story: "Three pieces of night"). In the second stage, the participants evaluated the gestures in the long video (i.e., the whole story).

Results of the short videos

The results from the ANOVA tests on the obtained data showed that: 1) the participants rated the temporal coordination between the robot gestures and the speech more positively when the gestures are produced in condition C1 (i.e., gesture were synchronized with speech) than when the gestures are produced in condition C2 (i.e., gestures were asynchronized with speech); 2) there was a signification effect between the participants' ratings for gestures in condition C1 and the participants' ratings for gestures in condition C2. This result supports the hypothesis H1. The gestures produced by our gesture scheduling mechanism appear synchronized with speech.

Similarly to the test of gesture timing, the results from the ANOVA tests showed that: 1) the participants rated the expressivity of the robot gestures more positively when the gestures are produced in condition C3 (i.e., gestures are produced with expressivity) than when the gestures are produced in condition C4 (i;e., gestures are produced without expressivity); 2) there was a significant effect

between the participants' ratings for the gestures in condition C3 and the participants' ratings for the gestures in condition C4. The hypothesis H2 is supported. Gestures modulated by the parameters temporal and spatial extents are perceived as expressive gestures.

However, the prediction in the hypothesis H4, H5 and H6 are not supported from the analysis of the ANOVA tests on the obtained results. The differences of the participants' perceptions via their ratings in condition C3 and in condition C4 are were not significant. The gesture expressivity and gesture timing did not have effect on the facts that the gestures and the speech convey complementary, contradictory or redundant information. Thus, the hypothesis H4, H5 and H6 have not been confirmed.

When we elaborated the gestures for the Nao robot, the shape of the gestures was based on the gestures of the real actors in the storytelling video corpus. Our expressive gesture model should reproduce human gestures for the robot in such a way that their meaning and shape are not changed from the original gestures. Hence, if the gestures done by the actors were complementary (respectively redundant or contradictory) with the speech, the robot gestures had to convey similar information. However, this conclusion needs to be verified by an analysis on the actors' gestures for the complementarity (respectively the redundancy or the contradictorily) of gestures and speech.

Because the Nao robot has physical constraints, certain gestures could not be completely reproduced from human gestures. Let us take as example two gestures shown in Figure 6.4: the French emblem gesture "I don't know" is made of a shrug and an arm movement with palm open up in front of speaker. However, the robot can execute the hand movements only. No shrug can be modeled. Similarly, in the French emblem gesture "I am hungry", two hands are pressed on the stomach, but the robot cannot do such a pressure correctly. When the robot does an incomplete gesture, it appears not to convey enough information to the participants. Additionally, we do not have a flexible tool to elaborate precise gestures for the robot. Thus, the shape of certain robot gestures loose part of their iconicity and as such become incomprehensible. Moreover, following Habets et al. (2011), the temporal coordination between gestures and speech also affects on the interpretation of the gestures. A gesture can be interpreted by multiple meanings by participants

in the absence of speech. The congruent interpretation of the gesture should be supported by its accompanying speech. When two modalities are not correctly synchronized, the gesture could be understood by a mismatched speech (i.e., by later or earlier words). We believe these reasons explain, at least partially, why the ratings of the participants did not validate the hypotheses H4, H5 and H6.

Results of the long video

For the participants' perception of the whole story, our analysis received positive results as shown in Figure 6.12: most of the participants found that the robot gestures were acceptable for two critical points: 76% participants agreed that the gestures were synchronized with speech and 70% participants agreed that the gestures were expressive while the robot was telling the story. Consequently, the two aspects of gestures (i.e., gesture timing and gesture expressivity) are validated for our expressive gesture model.

Regarding the statements that the gestures and the speech convey complementary, contradictory or redundant information, 60% participants slightly agreed that the robot's gestures and the speech convey complementary information. Most of them disagreed that the gestures and the speech were redundant or contradictory. These results can be explained by the same reasons exposed for the results of the short videos. The robot gestures had to convey similar information as the gestures done by the actors. But the lack of robot's gesture precision impairs on the readability of the gestures.

When comparing the results of the short videos with the results of the long video, the results of the long video are much higher than of the short videos. When the participants watch the whole story being told, they viewed the robot doing gestures with different expressivity. The could perceived the change in expressivity. On the contrary, in the short videos, the information conveyed by the robot gestures are always out of context as: 1) the order of the presentation of the short videos was selected randomly; 2) each video was one short segment of the story. We believe that the participants could made use of expressivity changes when evaluating the gesture quality in the long video.

That is why the average rating for the gestures in the short videos is lower than

the one in the long video.

6.8 Conclusion

The main findings based on human perceptions of the Nao storytelling robot are that: (1) participants in the experiment appreciated that robot gestures were synchronized with speech; (2) a majority of the participants considered robot gestures as being expressive; and (3) a majority of participants found robot gestures and speech convey complementary information. These experiment results validated that our framework generates communicative expressive gestures accompanying speech.

However, the results showed an important limitation of our system. The naturalness of robot gestures was not rated as acceptable in this experiment. Most of the participants disagreed that the robot gestures were natural. This problem came partly from certain physical constraints of a real robot and partly from our algorithm that did not fully restitute natural-like gestures in shape and timing. In particular, our system has not yet implemented certain gesture expressivity dimensions like the power, the tension and the fluidity of gestures. Additionally, the difference in velocity between stroke phase and other phases of a gesture has not yet studied in this model.

Chapter 7

Conclusion

7.1 Summary

This thesis presents a model to generate expressive communicative gestures accompanying speech for a humanoid agent. The model was designed in such a way that its processes are as much as possible independent of an agent's embodiment. So far, this model has been used to control gestures of the Greta virtual agents and the Nao physical humanoid robot.

The model is integrated within the GRETA multimodal behavior generation framework. While the gesture selection stage had been already implemented in the GRETA framework, my thesis work focuses on the gesture realization stage. This means that given selected gestures, our modules have to instantiate and schedule these gestures with a concrete action of hand-arm movements and then display them by an agent. The research work is related to studies of human gestures accompanying speech which are applied for a humanoid agent. Three research issues are addressed in this work. First, human gestures are encoded and reproduced in such a way that these gestures are realizable for agents. A set of properties of a gesture such as hand shape, wrist position, movement trajectory, etc is used in order to encode gestures. Second, gestures are planed to synchronize with speech. The model relies on the relationship between gestures and speech to schedule gesture timing. Third, these gestures are rendered expressively. Depending on the personality, the current emotional states of the agent, its gestures are varied by

modulating a set of gesture expressivity dimensions. The results of such research work were implemented in two modules of the GRETA framework: the first module is common to both agents, namely Behavior Realizer and the second module is specific to the Nao humanoid robot, namely Nao Animation Generator.

The first module, Behavior Realizer, deals with the three main aspects of gesture generation: gesture representation, gesture expressivity and gesture timing.

Regarding the gesture representation, we have proposed a gesture specification language to encode symbolic gesture templates into a gestuary. A gesture template containing only the stroke phase is used to rebuild the whole gesture on the fly. Because of the differences between the Greta agent and the Nao robot, each agent has its proper gestuary containing gestures suitable to its specification. A gestuary was built for the Nao robot based on gesture annotations extracted from real actors in a storytelling video corpus.

Concerning the gesture expressivity, the module uses a set of quality dimensions including spatial extent, temporal extent and stroke repetition to modulate the timing and the shape of the gesture trajectory. In this model, we modulate gestures on the fly using these parameters for both agents. From the same gesture template, the model produces gestures in different ways (e.g., slowly and narrowly vs. quickly and largely) depending on a combination of these expressivity parameters' values.

For the gesture timing, we worked on two tasks: 1) the simulation of human gesture velocity; and 2) the synchronization of gestures and speech. The first task was realized by using the Fitts' law to calculate the movement time of gestures. A robot may potentially need more time to execute hand movements than the time calculated by the Fitts' law. Thus, we had to pre-estimate minimum durations to execute robot gestures so that these gestures are scheduled correctly. The second task of the gesture-speech synchronization was ensured by adapting gesture movements to speech execution. The stroke timing of gestures is relative to speech timing via its time markers. Then the timing of other gesture phases is calculated from this stroke timing. Finally, gesture phases whose timing and forms are concretized are co-articulated to produce gesture trajectories.

The result of the Behavior Realizer module is a set of keyframes which stands for calculated gesture trajectories. The syntax of keyframes is common for both agents (i.e., Greta and Nao). The second module of Nao Animation Generator

receives keyframes as input and generates corresponding animation parameters to be played by the Nao robot.

The implemented model was evaluated through a set of perceptive tests on the Nao robot. We wanted to evaluate how robot gestures are perceived by human users at the level of the expressivity of gestures and the synchronization of gestures with speech while the robot was telling a story. Sixty three French speakers participated in our experiment. The obtained results showed that the co-verbal expressive gestures generated by our model and displayed by the Nao robot are acceptable. Forty eight participants (76%) agreed that gestures are synchronized with speech and forty four participants (70%) agreed that robot gestures are expressive.

7.2 Contributions

The main contribution of this thesis is the design, the implementation and the evaluation of a computational expressive communicative gesture engine integrated within the multimodal behavior generation GRETA system. Additionally, other contributes are:

1. This research work is an attempt towards a common multimodal generation framework for both virtual and physical agents. In our model, the similar characteristics between agents (i.e., Greta and Nao) are put into common modules and the differences between them are setup in external parameters (e.g., gestuary, velocity profile) or separated modules.

2. Our system is a SAIBA compliant ECA framework (Kopp et al., 2006). Like Greta, the international SAIBA framework was originally designed for virtual agents only (Kopp et al., 2006; Vilhjálmsson et al., 2007). The results of this thesis show an extensible capacity in controlling different embodiments of the SAIBA architecture.

3. We have defined an XML-based gesture specification language that is compatible with BML to describe behaviors in a gestuary. As the current version of BML focuses on temporal synchronization between modalities only, the

surface form of signals is open for development. Our work contributes to a gesture representation which is lacking in BML.

7.3 Suggestions for further work

Our expressive communicative gesture model for humanoid agents presents several limitations that we could improve in the future. One of the main limitations is to define gesture velocity profiles. Following Quek (1994)'s work, the velocity of wrist movements is not constant during a communicative gesture: the stroke phase is different from other phases in its velocity and in its acceleration. This means that we need to define one velocity profile for each gesture phase (i.e., preparation, stroke and retraction phases). This issue has not yet been implemented in our model.

Additionally, the issue of speed limit exists not only in robot but also in human subject. The reason is that the human body is a physical entity too. Thus, we need to define a maximal speed for gestures to be used to control virtual agent's movements. In our implementation, the Nao robot was pre-tested to find out thresholds for its joints' speed limit. However, the thresholds for Greta virtual agent have not yet been studied.

The system simulates human behaviors in realtime. Hence it must deal with continuous BML requests. A strategy to resolve the conflict between behaviors specified in these BML is necessary to be developed. For instance, if a new BML request arrives before the realization of previous requests has been completed, a BML composition mechanism has to indicate how the new behaviors specified in the new BML request collaborate with the behaviors in the current BML requests (e.g., replacing or merging or appending BML compositions).

To render expressive gestures, a set of gestural quality dimensions was applied to modulate gestures on the fly. However, certain quality dimensions such as Fluidity, Tension and Power have not yet been implemented for the Nao robot. The perception of human users on the quality of the robot gesture expressivity may be improved with the implementation of such dimensions.

As the gesture model is not limited to a specific virtual agent or a robot, we need to validate its performance on other humanoid agents. Additionally, we

should conduct a perceptive study to ensure that different agents being controlled by our model transmit similar information.

Appendix A

List of my publications

1. Q. A. Le, J.-F. Huang, C. Pelachaud, A Common Gesture and Speech Production Framework for Virtual and Physical Agents. In *Proceedings of the workshop on Speech and Gesture Production in Virtually and Physically Embodied Conversational Agents, the 14th ACM International Conference on Multimodal Interaction (ICMI2012)*, Oct 26, Santa Monica, CA, USA.

2. M. Ochs, E. Bevacqua, K. Prépin, Q. A. Le, Y. Ding, J.-F. Huang, R. Niewiadomski et C. Pelachaud, La compréhension de la machine a travers l'expression non-verbale, *III : Intercompréhension - de l'intraspécifique a l'interspécifique*, Novembre 2011 ,Nantes, France.

3. Q. A. Le, S. Hanoune and C. Pelachaud, Design and implementation of an expressive gesture model for a humanoid robot. In *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids2011)*, Oct 26-28, 2011, Bled, Slovenia.

4. Q. A. Le and C. Pelachaud, Expressive Gesture Model for Humanoid Robot. In *Proceedings of the fourth bi-annual International Conference of the HUMAINE Association on Affective Computing and Intelligent Interaction (ACII2011)*, Oct 9-12, 2011 Memphis, Tennessee, USA.

5. R. Niewiadomski, E. Bevacqua, Q. A. Le, M. Obaid, J. Looser, and C. Pelachaud, Cross-media agent platform. In *Proceedings of the 16th International Conference on 3D Web Technology (Web3D)*, 2011, Paris, France.

6. Q. A. Le and C. Pelachaud, Generating co-speech gestures for the humanoid robot NAO through BML. *In Lecture Notes in Computer Science (LNCS 2012). In Proceedings of the 9th International Gesture Workshop on Gesture in Embodied Communication and Human-Computer Interaction (GW2011)*, May 25-27, 2011, Athens, Greece.

7. Q. A. Le and C. Pelachaud, Expressive Gesture Model for Storytelling Humanoid Agent. In *Proceedings of the fourth workshop on Embodied Conversational Agents, (WACA2010)*, November 25-26, 2010, Lilles, France.

8. R. Gelin, C. D. Alessandro, O. Derroo, Q. A. Le, D. Doukhan, J.C. Martin, C. Pelachaud, A. Rilliard, S. Rosset. Tales Nao : Towards a storytelling humanoid robot. *Dialog with Robots 2010 AAAI Fall Symposium*, November 11-13, 2010 Arlington, VA, USA.

9. C. Pelachaud, R. Gelin, J.-C. Martin and Q. A. Le, Expressive Gestures Displayed by a Humanoid Robot during a Storytelling Application. In *Proceedings of the Second International Symposium on New Frontiers in Human-Robot Interaction (AISB 2010)*, 31 March Ű 1 April 2010, De Montfort University, Leicester, United Kingdom.

10. Q. A. Le, A. Popescu-Belis (2009). Automatic vs. human question answering over multimedia meeting recordings. In *10th Annual Conference of the International Speech Communication Association (InterSpeech2009)*, 6-10 September 2009, Brighton, United Kingdom.

11. Q.A. Le, A. Popescu-Belis (2008) - AutoBET: towards automatic answering of BET questions for meeting browser evaluation. Poster for the *IM2 Annual Review Meeting*, 12-13 November 2008, Martigny, Switzerland.

12. Q. A. Le, N. A. Nguyen. Some problems in extracting moving objects from video sequences. In *Young Vietnamese Scientists Meeting (YVSMŠ05)*, June 12-16, 2005, Nha Trang, Vietnam

13. Q. A. Le. T. L. Phan, H. L. Le, V. T. Nguyen. Applying Computer Vision to Traffic Surveillance in Vietnam. In *Sixth Vietnam Conference on Automation (VICA6)*, April 12-14, 2005, Hanoi, Vietnam

Bibliography

Allwood, J., Nivre, J., and Ahlsen, E. (1992). On the semantics and pragmatics of linguistic feedback. *Journal of semantics*, 9(1):1–26.

Argyle, M. (1988). *Bodily communication*. Methuen London Company.

Aronoff, J., Woike, B., and Hyman, L. (1992). Which are the stimuli in facial displays of anger and happiness? configurational bases of emotion recognition. *Journal of Personality and Social Psychology; Journal of Personality and Social Psychology*, 62(6):1050.

Bennewitz, M., Faber, F., Joho, D., and Behnke, S. (2007). Fritz-a humanoid communication robot. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 1072–1077. IEEE.

Bergmann, K., Aksu, V., and Kopp, S. (2011). The relation of speech and gestures: Temporal synchrony follows semantic synchrony. *Proceedings of the 2nd Workshop on Gesture and Speech in Interaction (GeSpIn 2011)*.

Bergmann, K. and Kopp, S. (2009). Gnetic–using bayesian decision networks for iconic gesture generation. In *Intelligent Virtual Agents*, pages 76–89. Springer.

Bergmann, K. and Kopp, S. (2010). Modeling the production of coverbal iconic gestures by learning bayesian decision networks. *Appl. Artif. Intell.*, 24(6):530–551.

Bevacqua, E. (2009). *Computational model of listener behavior for Embodied Conversational Agents*. Phd thesis, University of Paris 8.

Bevacqua, E., Mancini, M., Niewiadomski, R., and Pelachaud, C. (2007). An expressive eca showing complex emotions. In *Proceedings of the AISB annual convention, Newcastle, UK*, pages 208–216. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Bevacqua, E. and Pelachaud, C. (2004). Expressive audio-visual speech. *Computer Animation and Virtual Worlds*, 15(3-4):297–304.

Bevacqua, E., Prepin, K., Niewiadomski, R., Sevin, E., and Pelachaud, C. (2010). Greta: Towards an interactive conversational virtual companion. *Close Engagements with Artificial Companions: Key social, psychological, ethical and deisgn issues*, pages 143–156.

Bitti, P. and Poggi, I. (1991). 12. symbolic nonverbal behavior: Talking through gestures. *Fundamentals of nonverbal behavior*, page 433.

Black, A., Taylor, P., and Caley, R. (1998). The festival speech synthesis system. *University of Edinburgh*.

Boone, R. and Cunningham, J. (1996). ChildrenŠs understanding of emotional meaning in expressive body movement. In *Biennial Meeting of the Society for Research in Child Development*.

Breazeal, C. (2003). Toward sociable robots. *Robotics and Autonomous Systems*, 42(3):167–175.

Breazeal, C., Kidd, C., Thomaz, A., Hoffman, G., and Berlin, M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 708–713. IEEE.

Bremner, P., Pipe, A., Melhuish, C., Fraser, M., and Subramanian, S. (2009). Conversational gestures in human-robot interaction. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 1645–1649. IEEE.

Buisine, S., Hartmann, B., Mancini, M., and Pelachaud, C. (2006). Conception et évaluation d'un modèle d'expressivité pour les gestes des agents conversationnels. *Revue d'Intelligence Artificielle*, pages 621–638.

Butcher, C. and Goldin-Meadow, S. (2000). Gesture and the transition from one-to two-word speech: When hand and mouth come together. *Language and gesture*, 2:235–257.

Butterworth, B. and Hadar, U. (1989). Gesture, speech, and computational stages: A reply to mcneill. *American Psychological Association*.

Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjálmsson, H., and Yan, H. (1999). Embodiment in conversational interfaces: Rea. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 520–527. ACM.

Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., and Stone, M. (1994). Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 413–420. ACM.

Cassell, J., Vilhjálmsson, H., and Bickmore, T. (2001). Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486. ACM.

Chafai N.E., P. C. P. D. (2007). A semantic description of gesture in BML. In *Proceedings of AISB'07 Annual Convention Workshop on Language, Speech and Gesture for Expressive Characters, Newcastle, UK (2007)*.

Chi, D., Costa, M., Zhao, L., and Badler, N. (2000). The EMOTE model for effort and shape. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 173–182. ACM Press/Addison-Wesley Publishing Co.

Dawson, H. (1967). Basic human anatomy. *The American Journal of the Medical Sciences*, 254(1):123.

de Melo, C. and Paiva, A. (2008). Modeling gesticulation expression in virtual humans. *New Advances in Virtual Humans*, pages 133–151.

De Ruiter, J. (2000). The production of gesture and speech. *Language and gesture*, 2:284–311.

De Ruiter, J. P. (1998). *Gesture and Speech Production*. Doctoral dissertation at Catholic University of Nijmegen, Netherlands.

DeCarolis, B., Pelachaud, C., Poggi, I., and Steedman, M. (2004). {APML}, a Mark-up Language for Believable Behavior Generation. In Prendinger, H. and Ishizuka, M., editors, *Life-like Characters. {T}ools, Affective Functions and Applications*, pages 65–85. Springer.

Doukhan, D., Rilliard, A., Rosset, S., Adda-Decker, M., and dŠAlessandro, C. (2011). Prosodic analysis of a corpus of tales. In *Twelfth Annual Conference of the International Speech Communication Association*.

Efron, D. (1941). Gesture and environment (Book). *Oxford, England: King'S Crown Press*, page 184.

Ekman, P. and Friesen, W. (1972). Hand movements. *Journal of communication*, 22(4):353–374.

Ekman, P. and Friesen, W. (1981). The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Nonverbal communication, interaction, and gesture*, pages 57–106.

Faber, F., Bennewitz, M., Eppner, C., Gorog, A., Gonsior, C., Joho, D., Schreiber, M., and Behnke, S. (2009). The humanoid museum tour guide robotinho. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 891–896. IEEE.

Ferré, G. (2010). Relations temporelles entre parole et gestualité co-verbale en français spontané. *Actes des Journées d'Etude sur la Parole 2010*, pages 13–16.

Fitts, P. M. (1992). The information capacity of the human motor system in controlling the amplitude of movement. 1954. *Journal of experimental psychology. General*, 121(3):262–9.

Gallagher, H. and Frith, C. (2004). Dissociable neural pathways for the perception and recognition of expressive and instrumental gestures. *Neuropsychologia*, 42(13):1725–1736.

Gallaher, P. E. (1992). Individual differences in nonverbal behavior: Dimensions of style. *Journal of Personality and Social Psychology*, 63(1):133–145.

Gibet, S., Lebourque, T., and Marteau, P.-F. (2001). High-level Specification and Animation of Communicative Gestures. *Journal of Visual Languages & Computing*, 12(6):657–687.

Goldin-Meadow, S. (1999). The role of gesture in communication and thinking. *Trends in cognitive sciences*, 3(11):419–429.

Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2008). The nao humanoid: a combination of performance and affordability. *IEEE Transactions on Robotics*.

Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of nao humanoid. *The Int. Conf. on Robotics and Automation, 2009.*, pages 769–774.

Habets, B., Kita, S., Shao, Z., Özyurek, A., and Hagoort, P. (2011). The role of synchrony and ambiguity in speech–gesture integration during comprehension. *Journal of Cognitive Neuroscience*, 23(8):1845–1854.

Haring, M., Bee, N., and Andre, E. (2011). Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots. In *RO-MAN, 2011 IEEE*, pages 204–209.

Hartmann, B., Mancini, M., Buisine, S., and Pelachaud, C. (2005a). Design and evaluation of expressive gesture synthesis for embodied conversational agents. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1095–1096. ACM.

Hartmann, B., Mancini, M., and Pelachaud, C. (2005b). Towards affective agent action: Modelling expressive eca gestures. In *International conference on Intelligent User Interfaces-Workshop on Affective Interaction, San Diego, CA*.

Hartmann, B., Mancini, M., and Pelachaud, C. (2006). Implementing expressive gesture synthesis for embodied conversational agents. *LNCS: Gesture in human-Computer Interaction and Simulation*, pages 188–199.

Heloir, A. and Kipp, M. (2009). EMBR: A realtime animation engine for interactive embodied agents. *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 1:1–2.

Heloir, A. and Kipp, M. (2010a). Real-time animation of interactive agents: Specification and realization. *Applied Artificial Intelligence*, 24(6):510–529.

Heloir, A. and Kipp, M. (2010b). Requirements for a gesture specification language. *Gesture in Embodied Communication and Human-Computer Interaction*, pages 207–218.

Heylen, D., Kopp, S., Marsella, S. C., Pelachaud, C., and Vilhjálmsson, H. (2008). The Next Step towards a Function Markup Language. In *Proceedings of the 8th international conference on Intelligent Virtual Agents*, IVA '08, pages 270–280, Berlin, Heidelberg. Springer-Verlag.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1321–1326. IEEE.

Hogrefe, K., Ziegler, W., Weidinger, N., and Goldenberg, G. (2011). Non-verbal communication in severe aphasia: Influence of aphasia, apraxia, or semantic processing? *Cortex*.

Holroyd, A. and Rich, C. (2012). Using the behavior markup language for human-robot interaction. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*, pages 147–148.

Holroyd, A., Rich, C., Sidner, C., and Ponsler, B. (2011). Generating connection events for human-robot collaboration. In *RO-MAN, 2011 IEEE*, pages 241–246. IEEE.

Holz, T., Dragone, M., and O?Hare, G. M. P. (2009). Where robots and virtual agents meet. *International Journal of Social Robotics*, 1(1):83–93.

Huang, J. and Pelachaud, C. (2012). An efficient energy transfer inverse kinematics solution. *Motion In Game*.

Iverson, J. M., Goldin-Meadow, S. (1998). Why people gesture when they speak. *Nature*, 396(November):228.

Kelly, S., Barr, D., Church, R., and Lynch, K. (1999). Offering a hand to pragmatic understanding: The role of speech and gesture in comprehension and memory. *Journal of Memory and Language*, 40(4):577–592.

Kendon, A. (1972). Some relationships between body motion and speech. *Studies in dyadic communication*, pages 177–210.

Kendon, A. (1980). Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25:207–227.

Kendon, A. (1994). Do gestures communicate? A review. *Research on language and social interaction*, 27(3):175–200.

Kendon, A. (2004). *Gesture: Visible Action as Utterance*.

Kipp, M. (2005). *Gesture generation by imitation: From human behavior to computer character animation*. Dissertation.

Kipp, M. et al. (2008). Spatiotemporal coding in anvil. In *Proceedings of the 6th international conference on Language Resources and Evaluation (LREC-08)*.

Kita, S., Van Gijn, I., and der Hulst, H. (1998). Movement phases in signs and co-speech gestures, and their transcription by human coders. *Gesture and sign language in human-computer interaction*, pages 23–35.

Kopp, B. S., Wachsmuth, I., and Kopp, S. (2004a). Synthesizing multimodal utterances for conversational agents: Research Articles. *Comput. Animat. Virtual Worlds*, 15(1):39–52.

Kopp, S. (2005). Surface realization of multimodal output from xml representations in murml. In *Invited Workshop on Representations for Multimodal Generation*.

Kopp, S., Bergmann, K., and Wachsmuth, I. (2008). Multimodal communication from multimodal thinking towards an integrated model of speech and gesture production. *International Journal of Semantic Computing*, 2(01):115–136.

Kopp, S., Jung, B., Lessmann, N., and Wachsmuth, I. (2003). Max - A Multimodal Assistant in Virtual Reality Construction. *KI*, 17(4):11.

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 *Intelligent Virtual Agents*, pages 205–217. Springer.

Kopp, S., Tepper, P., and Cassell, J. (2004b). Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 97–104. ACM.

Kopp, S. and Wachsmuth, I. (2002). Model-based Animation of Coverbal Gesture. In *Proceedings of the Computer Animation*, CA '02, pages 252—-, Washington, DC, USA. IEEE Computer Society.

Krauss, R., Chen, Y., and Gotfexnum, R. (2000). 13 Lexical gestures and lexical access: a process model. *Language and gesture*, pages 261–283.

Krauss, R. and Hadar, U. (1999). The role of speech-related arm/hand gestures in word retrieval. *Gesture, speech, and sign*, pages 93–116.

Lee, J. and Marsella, S. (2006). Nonverbal behavior generator for embodied conversational agents. In *Proceedings of the 6th international conference on Intelligent Virtual Agents*, IVA'06, pages 243–255, Berlin, Heidelberg. Springer-Verlag.

Levelt, W. (1989). *Speaking: From intention to articulation.* Cambridge, MA, The MIT Press.

Levelt, W. (1993). Lexical access in speech production. *Knowledge and language: From Orwell's problem to Plato's problem*, 241.

Levelt, W. (1996). A theory of lexical access in speech production. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 3–3. Association for Computational Linguistics.

Levelt, W., Schriefers, H., Vorberg, D., Meyer, A., Pechmann, T., and Havinga, J. (1991). The time course of lexical access in speech production: A study of picture naming. *Psychological review*, 98(1):122.

Loyall, A. and Bates, J. (1997). Personality-rich believable agents that use language. In *Proceedings of the first international conference on Autonomous agents*, pages 106–113. ACM.

MacKenzie, I. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139.

Mancini, M. (2008). *Multimodal distinctive behavior for expressive embodied conversational agents.* Phd thesis, University of Paris 8.

Mancini, M., Castellano, G., Peters, C., and McOwan, P. (2011). Evaluating the communication of emotion via expressive gesture copying behaviour in an embodied humanoid agent. *Affective Computing and Intelligent Interaction*, pages 215–224.

Mancini, M., Niewiadomski, R., Bevacqua, E., and Pelachaud, C. (2008). Greta: a saiba compliant eca system. In *Troisiéme Workshop sur les Agents Conversationnels Animés.*

Mancini, M. and Pelachaud, C. (2007). Dynamic behavior qualifiers for conversational agents. *Intelligent Virtual Agents*, pages 112–124.

Mancini, M. and Pelachaud, C. (2008a). Distinctiveness in multimodal behaviors. In *Proceedings of Conference on Autonomous Agents and Multi-Agent Systems (AAMAS08)*.

Mancini, M. and Pelachaud, C. (2008b). The fml-apml language. *Why Conversational Agents do what they do. Functional Representations for Generating Conversational Agent Behavior. AAMAS*.

Manohar, V., al Marzooqi, S., and Crandall, J. W. (2011). Expressing emotions through robots: a case study using off-the-shelf programming interfaces. In *The 6th Int. Conf. on HRI*, pages 199–200. ACM.

Martin, J.-C. (2009). The contact video corpus.

Mataric, M. (2000). Getting humanoids to move and imitate. *Intelligent Systems and Their Applications, IEEE*, 15(4):18–24.

Mataric, M., Williamson, M., Demiris, J., and Mohan, A. (1998). Behavior-based primitives for articulated control. In *R. Pfiefer, B. Blumberg, J.-AM. SWW (Ed.), Fifth International conference on simulation of adaptive behavior SAB*, volume 98, pages 165–170.

McNeill, D. (1985). So you think gestures are nonverbal? *Psychological review*, 92(3):350.

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought*. University of Chicago Press.

McNeill, D., Duncan, S., et al. (2000). Growth points in thinking-for-speaking. *Language and gesture*, pages 141–161.

Mead, R., Wade, E., Johnson, P., Clair, A., Chen, S., and Mataric, M. (2010). An architecture for rehabilitation task practice in socially assistive human-robot interaction. In *19th IEEE International Symposium in Robot and Human Interactive Communication*.

Meijer, M. (1989). The contribution of general features of body movement to the attribution of emotions. *Journal of Nonverbal Behavior*, 13(4):247–268.

Monceaux, J., Las, E., Lemoine, C., and Mazel, A. (2011). Demonstration Ũ First Steps in Emotional Expression of the Humanoid Robot Nao. pages 235–236.

Montepare, J., Koff, E., Zaitchik, D., and Albert, M. (1999). The use of body movements and gestures as cues to emotions in younger and older adults. *Journal of Nonverbal Behavior*, 23(2):133–152.

Morrel-Samuels, P. and Krauss, R. (1992). Word familiarity predicts temporal asynchrony of hand gestures and speech. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(3):615.

Neff, M. and Fiume, E. (2006). Methods for exploring expressive stance. *Graphical Models*, 68(2):133–157.

Neff, M., Kipp, M., Albrecht, I., and Seidel, H. (2008). Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics (TOG)*, 27(1):5.

Ng-thow hing, V. and Okita, S. (2010). Synchronized Gesture and Speech Production for Humanoid Robots. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4617–4624.

Niewiadomski, R., Bevacqua, E., Mancini, M., and Pelachaud, C. (2009). Greta: an interactive expressive ECA system. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1399–1400, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Niewiadomski, R., Huang, J., and Pelachaud, C. (2012). Effect of facial cues on identification. *25th Annual Conference on Computer Animation and Social Agents (CASA 2012), Singapore, May 2012*.

Nozawa, Y., Dohi, H., Iba, H., and Ishizuka, M. (2004). Humanoid robot presentation controlled by multimodal presentation markup language mpml. *Computer animation and virtual worlds*, pages 153–158.

OxfordDictionary (2012). Oxford english dictionary. Online.

Pelachaud, C. (2005). Multimodal expressive embodied conversational agents. *The 13th annual ACM Int. Conf. on Multimedia*, pages 683–689.

Poggi, I. (2002). From a typology of gestures to a procedure for gesture production. *Gesture and sign language in human-computer interaction*, pages 158–168.

Poggi, I. (2008). Iconicity in different types of gestures. *Special issue of Gesture*, 8(1):45–61.

Poggi, I. and Pelachaud, C. (2008). Persuasion and the expressivity of gestures in humans and machines. *Embodied Communication in Humans and Machines*, pages 391–424.

Poggi, I., Pelachaud, C., and Caldognetto, E. (2004). Gestural mind markers in ecas. *Gesture-Based Communication in Human-Computer Interaction*, pages 481–482.

Pot, E., Monceaux, J., Gelin, R., and Maisonnier, B. (2009). Choregraphe: a graphical tool for humanoid robot programming. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 46–51. IEEE.

Prillwitz, S. (1989). *HamNoSys Version 2.0: Hamburg notation system for sign languages: An introductory guide*. Signum.

Quek, F. (1994). Toward a vision-based hand gesture interface. In *Proceedings of the conference on Virtual reality software and technology*, pages 17–31. World Scientific Publishing Co., Inc.

Quek, F. (1995). Eyes in the interface. *Image and Vision Computing*, 13(6):511–525.

Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X., Kirbas, C., McCullough, K., and Ansari, R. (2002). Multimodal human discourse: gesture and speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(3):171–193.

R. Gelin, C. d'Alessandro, O. Derroo, Q.A. Le, D. Doukhan, J.C. Martin, C. Pelachaud, A. Rilliard, S. R. (2010). Towards a Storytelling Humanoid Robot.

Dialog with Robots Ũ 2010 AAAI Fall Symposium, November 11-13, 2010 Arlington, VA, USA, pages 137–138.

Reidsma, D., de Kok, I., Neiberg, D., Pammi, S., van Straalen, B., Truong, K., and Van Welbergen, H. (2011). Continuous interaction with a virtual human. *Journal on Multimodal User Interfaces*, 4(2):97–118.

Rimé, B. and Schiaratura, L. (1991). *Gesture and speech.* New York, NY, US: Cambridge University Press; Paris, France: Editions de la Maison des Sciences de l'Homme.

Ruttkay, Z., Pelachaud, C., Poggi, I., and Noot, H. (2008). Exercises in style for virtual humans. *Animating expressive characters for social interaction*, 143.

Salem, M. (2012). *Conceptual Motorics-Generation and Evaluation of Communicative Robot Gesture.* Phd thesis, Bielefeld University.

Salem, M., Kopp, S., Wachsmuth, I., and Joublin, F. (2010a). Generating robot gesture using a virtual agent framework. *Intelligent Robots and Systems (IROS 2010)*, pages 3592–3597.

Salem, M., Kopp, S., Wachsmuth, I., and Joublin, F. (2010b). Towards an integrated model of speech and gesture production for multi-modal robot behavior. pages 649–654.

Salem, M., Kopp, S., Wachsmuth, I., Rohlfing, K., and Joublin, F. (2012). Generation and evaluation of communicative robot gesture. *International Journal of Social Robotics*, pages 1–17.

Schmitz, C. (2010). Limesurvey (computer software).

Schröder, M., Hunecke, A., and Krstulovic, S. (2006). Openmary–open source unit selection as the basis for research on expressive synthesis. In *Proc. Blizzard Challenge*, volume 6.

Shi, C., Kanda, T., Shimada, M., Yamaoka, F., Ishiguro, H., and Hagita, N. (2010). Easy development of communicative behaviors in social robots. In *In-*

telligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 5302–5309. IEEE.

Shiomi, M., Kanda, T., Ishiguro, H., and Hagita, N. (2006). Interactive humanoid robots for a science museum. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 305–312. ACM.

Snyder, B., Bosnanac, D., and Davies, R. (2011). *ActiveMQ in action*. Manning Publications Company.

Stone, M., DeCarlo, D., Oh, I., Rodriguez, C., Stere, A., Lees, A., and Bregler, C. (2004). Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (TOG)*, 23(3):506–513.

Sugiyama, O., Kanda, T., Imai, M., Ishiguro, H., and Hagita, N. (2007). Natural deictic communication with humanoid robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1441–1448. IEEE.

Thiebaux, M., Marsella, S., Marshall, A., and Kallmann, M. (2008a). Smartbody: Behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 151–158. International Foundation for Autonomous Agents and Multiagent Systems.

Thiebaux, M., Rey, M., Marshall, A. N., Marsella, S., and Kallmann, M. (2008b). SmartBody: behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1*, number Aamas in AAMAS '08, pages 151–158, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Van Welbergen, H., Nijholt, A., Reidsma, D., and Zwiers, J. (2005). Presenting in virtual worlds: Towards an architecture for a 3d presenter explaining 2d-presented information. *Intelligent Technologies for Interactive Entertainment*, pages 203–212.

Van Welbergen, H., Reidsma, D., and Kopp, S. (2012). An incremental multimodal realizer for behavior co-articulation and coordination. In *Intelligent Virtual Agents*, pages 175–188. Springer.

Vilhjálmsson, H., Cantelmo, N., Cassell, J., E. Chafai, N., Kipp, M., Kopp, S., Mancini, M., Marsella, S., Marshall, A. N., Pelachaud, C., Ruttkay, Z., Thórisson, K. R., Welbergen, H., and Werf, R. J. (2007). The Behavior Markup Language: Recent Developments and Challenges. In *Proceedings of the 7th international conference on Intelligent Virtual Agents*, IVA '07, pages 99–111, Berlin, Heidelberg. Springer-Verlag.

Wallbott, H. (1998). Bodily expression of emotion. *European journal of social psychology*, 28(6):879–896.

Wallbott, H., Scherer, K., et al. (1986). Cues and channels in emotion recognition. *Journal of Personality and Social Psychology; Journal of Personality and Social Psychology*, 51(4):690.

Wallbott, H. G. (1985). Hand movement quality: a neglected aspect of nonverbal behavior in clinical judgment and person perception. *Journal of Clinical Psychology*, 41(3):345–359.

Website (2011). Bml 1.0 standard. *http://www.mindmakers.org/projects/bml-1-0/wiki*.

Welbergen, H., Reidsma, D., Ruttkay, Z. M., and Zwiers, J. (2010). Elckerlyc. *Journal on Multimodal User Interfaces*, 3(4):271–284.

Xing, S. and Chen, I. (2002). Design expressive behaviors for robotic puppet. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 1, pages 378–383. IEEE.