Generating co-speech gestures for the humanoid robot NAO through BML

Le Quoc Anh and Catherine Pelachaud

CNRS, LTCI Telecom ParisTech, France
Emails: {quoc-anh.le, catherine.pelachaud}@telecom-paristech.fr

Abstract. We develop an expressive gesture model based on GRETA platform to generate gestures accompanying speech for different embodiments. This paper presents our ongoing work on an implementation of this model for the humanoid robot NAO. From a specification of multimodal behaviors encoded with the behavior markup language, BML, the system synchronizes and realizes the verbal and nonverbal behaviors on the robot.

Keywords: Humanoid robot, expressive gesture, Nao, Greta, BML, SAIBA

1 Introduction

We aim at building a model generating expressive communicative gestures for embodied agents such as the humanoid robot Nao [2] and the virtual agent Greta [11]. To reach this goal, we extend and develop our existing virtual agent system GRETA [11], which follows the SAIBA framework (cf. Figure 1). GRETA consists of three separated modules: the first module, Intent Planning, defines communicative intents to be conveyed. The second, Behavior Planning, plans the corresponding multimodal behaviors to be realized, and the third module, Behavior Realizer, synchronizes and realizes the planned behaviors.

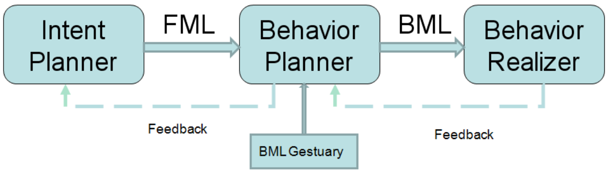


Fig. 1. The SAIBA framework for multimodal behavior generation [6].

The results of the first module is the input of the second module through an interface described with a representation markup language, named FML (Function Markup Language). The output of the second module is encoded with another representation language, named BML [6] and then sent to the third module.

Both FML and BML are XML languages and they do not refer to any particular parameters of an agent (e.g. wrist joint).

From any given communicative intentions, the system selects and plans gestures from a repository, called Gestural Lexicon or Gestuary (cf. Figure 1). These gestures are described symbolically with an extension of the gesture representation language BML. In SAIBA framework, the lexicon is supposed to be player-independent. However, our model uses the behavior library in a way that it provides not only a means to combine multiple behavior signals for any given communicative intention, but also supports to specify constraints to do behaviors such as the limit of movement space and speed for gestures. That means both Behavior Planner and Behavior Realizer modules need access to the behavior library. At the Behavior Planner, the lexicon gives a list of available behaviors and some constraints between them when they are combined. Meanwhile, at the stage of the Behavior Realizer, it provides more detailed constraints of each behavior signal (e.g. gesture) to be realized.

We want to be able to use the same system to control both agents (i.e. the virtual one and the physique one). However, the robot and the agent do not have the same behavior capacities (e.g. the robot can move its legs and torso but does not have facial expression and has very limited arm movements). Therefore the nonverbal behaviors to be displayed by the robot should be different from those of the virtual agent. For instance, the robot has only two hand configurations, open and closed; it cannot extend one finger only. 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 the robot and the virtual agent, while taking into account the physical constraint of both, we consider two repertoires of gestures, one for the robot and the other 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. The elaboration of repertoires encompasses the notion of *gesture family with variants* proposed by Calbris [1]. Gestures from the same family convey similar meanings but may differ in their shape (i.e. the element *deictic* exists in both repertoires; it corresponds to an extended finger or to an arm extension). In the proposed model, therefore the Behavior Planning module remains the same for them and unchanged from the GRETA system. From the BML file outputted by the Behavior Planner, we instantiate the BML tags from either gesture repertoires. That is, given a set of intentions and emotions to convey, GRETA computes, through the Behavior Planning, the corresponding sequence of behaviors specified with BML. The Behavior Realizer module has been developed to create the animation adaptable to the agents, who have different behavior capabilities. The Figure 2 presents an overview of our system.

In this paper, we presents our design and implementation of the expressive gesture model for the humanoid robot NAO. This work is conducted within the frame of the French Nation Agency for Research project, namely GVLEX, whose objective is to build an expressive robot able to display communicative gestures

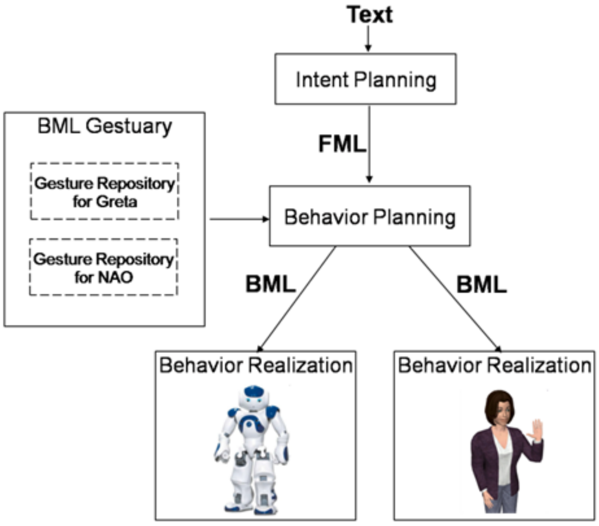


Fig. 2. System Overview.

with different behavior qualities while telling a story. While other partners of the project deal with expressive voice, our work focuses on expressive nonverbal behaviors, especially on gestures. In this project, we have elaborated a repository of gestures specific to the robot using gesture annotations extracted from a storytelling video corpus [8]. The model takes into account the physical characteristics of the robot. Each gesture is guaranteed to be executable by the robot. When gestures are realized, their expressivity is increased by considering a set of quality dimensions such as the amplitude (SPC), fluidity (FLD), power (PWR), or speed of gestures (TMP) that has been previously developed for the virtual agent Greta [3].

The paper is structured as follows. The next section describes some recent initiatives in controlling humanoid robot gestures. Then, Section 3 presents in details the design and implementation of a gesture database and a behavior realizer for the robot. Section 4 makes conclusions and proposes some future works.

2 State of the art

Several initiatives have been proposed recently to control multimodal behaviors of a humanoid robot. Salem et al. [15] use a gesture engine of the virtual agent Max to drive the humanoid robot ASIMO. Rich et al. [14] implement a system following an event-driven architecture to solve the problem of unpredictability in performance of their humanoid robot Melvin. Ng-Thow-Hing et al. [10] develop a system that takes any text and then selects and produces the corresponding gestures to be performed by the robot ASIMO. Kushida et al.[7] equip their robot with a capacity of producing deictic gestures when the robot gives a presentation on the screen. These systems have several common characteristics. They calculate animation parameters of the robot from a symbolic description encoded with a script language such as BML [14], MURML [15], MPML-HR [7], etc. The synchronisation of gestures with speech is guaranteed by adapting the gesture movements to the structure of speech [15, 10]. This is also the method used in our system. Some systems have a feedback mechanism to receive and process feedback from the robot in real time. The feedback information is used to improve the gesture movements [15], or synchronize gestures with speech [14].

Our system has some differences from the others. It focuses not only on the coordination of gestures and speech but also on the signification and the expressivity of gestures on the robot. While the gesture signification is studied carefully when elaborating a repertoire of robot gestures, the gesture expressivity is increased by adding gesture dimension parameters such as spatial extension (SPC), temporal extension (TMP) when creating gesture animation for the robot.

3 System Design and Implementation

The proposed model is developed based on the GRETA framework. It uses the existing Behavior Planner module of the GRETA system to select and plan multimodal behaviors. A new Behavior Realizer module has been developed to adapt the behavior capabilities of the robot. The main objective of this module is to generate animation, which will be displayed by the robot from the received BML message. This process is divided into two tasks: the first one is to create a gesture database specific to the robot and the second one is to build a robot speech-gesture production engine. Figure 3 gives an outline of the system that will be presented in detail in the following subsections.

3.1 Gesture database

The robot has physical constraints such as the limit of the movement speed and space. Hence the hand-arm movement speed and space specifications should be determined when building gestures. Each gesture of the lexicon is assured to be engaged with these specifications. The elaboration of gestures starts from gesture annotations extracted from a video corpus. Gesture prototypes are formed and stored symbolically in a repository of gestures (i.e. BML gestuary or gesture lexicon). All gesture lexicon are tested to guarantee its realizability on the robot (e.g. avoid collision or conflict between robot joints when doing a gesture, or void singular positions where the robot hand cannot reach). The description of the gestures are symbolic so that they can be instantiated dynamically into joint values of the robot when creating the animation.

Gesture annotations The elaboration of symbolic gestures in the robot lexicon is based on gesture annotations extracted from a Storytelling Video Corpus. The video corpus was recorded and annotated by Jean-Claude Martin et al.

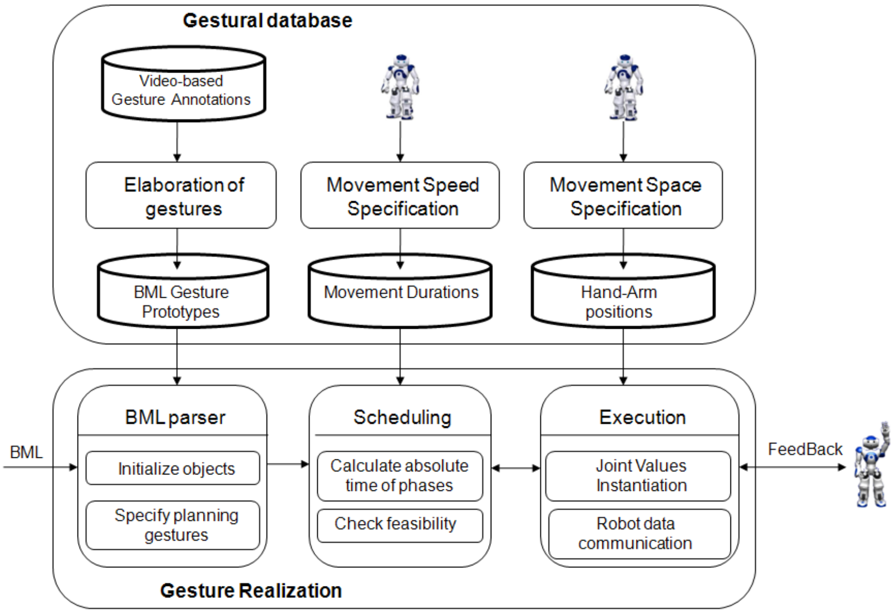
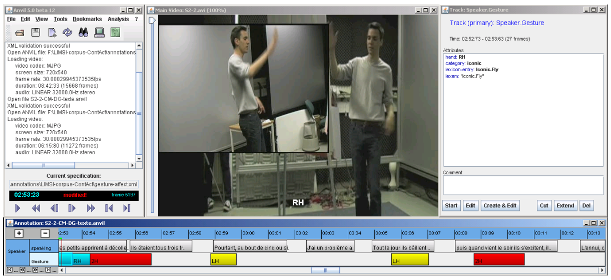


Fig. 3. Outline of the system.

[8], a partner of the GVLEX project. To create this corpus, six actors were videotaped while telling a French story "Three Little Pieces of Night" twice. Two cameras were used (front and side view) to get postural expressions in the three dimensions space. Then, the Anvil video annotation tool [5] is used to annotate gesture information (cf. Figure 4). Each gesture of the actors is annotated with information of its category (i.e. iconic, beat, metaphoric and deictic), its duration and which hand is being used, etc. From the form of gestures displayed on the video with their annotated information, we have elaborated the symbolic gestures correspondingly. These gestures are encoded using a set of gesture specifications that will be presented in the next section.



```
XML validation successful
Open ANVIL file: F:\LIMSI-corpus-ContAct\annotations
Loading video:
    video codec: MJPG
    screen size: 720x540
    frame rate: 30.00029945373535fps
    duration: 08:42:33 (15668 frames)
    audio: LINEAR 32000.0Hz stereo
Open file S2-2-CM-DG-texte.anvil
XML validation successful
Open ANVIL file: F:\LIMSI-corpus-ContAct\annotations
Loading video:
    video codec: MJPG
    screen size: 720x540
    frame rate: 30.00029945373535fps
    duration: 06:15:80 (11272 frames)
    audio: LINEAR 32000.0Hz stereo
```

Current specification: .annotations\LIMSI-corpus-ContAct\gesture-affect.xml

02:53:23 modified! frame 5197

Track (primary): Speaker.Gesture

Time: 02:52:73 - 02:53:63 (27 frames)

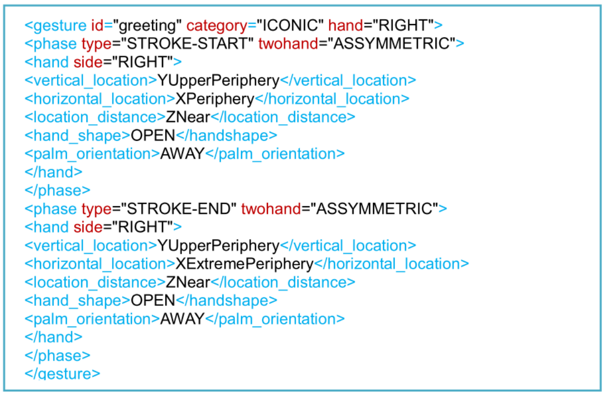
Attributes
hand: RH
category: iconic
lexicon-entry: Iconic.Fly
lexem: "Iconic.Fly"

Comment

es petits apprirent à décolle.. Ils étaient tous trois tr.. Pourtant, au bout de cinq ou s.. J'ai un problème a.. Tout le jour ils bâillent .. puis quand vient le soir ils s'excitent, il.. L'ennui, c

Fig. 4. Gestura annotations with Anvil tool.

Gesture Specification We have proposed a new XML schema as an extension of BML language to describe symbolically gestures in gesture repositories (i.e. lexicons). The specification of a gesture relies on the gesture description of McNeill [9], the gesture hierarchy of Kendon [4] and some notions from the HamNoSys system [13]. As a result, a gesture action may be divided into several phases of wrist movements, in which the obligatory phase is called stroke carrying the meaning of the gesture. The stroke may be preceded by a preparatory phase, which takes the articulatory joints (i.e. hands and wrists) to the position ready for the stroke phase. After that, it may be followed by a retraction phase that returns the hands and arms of the agent to the relax position or a position initialized for the next gesture (cf. Figure 11). In the lexicon, only the description of stroke phase is specified for each gesture. Other phases are generated automatically by the system. A stroke phase is represented through a sequence of key poses, each of which is described with the information of hand shape, wrist position, palm orientation, etc. The wrist position is always defined by three tags *verticallocation* that corresponds to the Y axis, *horizontallocation* that corresponds to the X axis, and *locationdistance* corresponding to the Z axis in a limited movement space.



```
<gesture id="greeting" category="ICONIC" hand="RIGHT">
<phase type="STROKE-START" twohand="ASSYMMETRIC">
<hand side="RIGHT">
<vertical_location>YUpperPeriphery</vertical_location>
<horizontal_location>XPeriphery</horizontal_location>
<location_distance>ZNear</location_distance>
<hand_shape>OPEN</handshape>
<palm_orientation>AWAY</palm_orientation>
</hand>
</phase>
<phase type="STROKE-END" twohand="ASSYMMETRIC">
<hand side="RIGHT">
<vertical_location>YUpperPeriphery</vertical_location>
<horizontal_location>XExtremePeriphery</horizontal_location>
<location_distance>ZNear</location_distance>
<hand_shape>OPEN</handshape>
<palm_orientation>AWAY</palm_orientation>
</hand>
</phase>
</gesture>
```

Fig. 5. An example of gesture specification.

Following the gesture space proposed by McNeill [9], we have five horizontal values (XEP, XP, XC, XCC, XOppC), seven vertical values (YUpperEP, YUpperP, YUpperC, YCC, YLowerC, YLowerP, YLowerEP), and three distance values (Znear, Zmiddle, Zfar) as illustrated in Figure 6. By combining these values, we have 105 possible wrist positions. An example of the description for the greeting gesture is presented in Figure 5. In this gesture, the stroke phase consists of two key poses. These key poses represent the position of the right hand (i.e. above the head), the hand shape (i.e. open) and the palm orientation (i.e. foward). Two key poses are different from only one symbolic value of horizontal position. This is to display a wave hand movement when greeting someone. The NAO robot cannot rotate its wrist (i.e. it has only the WristYaw joint). Consequently, there is no description of wrist orientation in the gesture specification for the robot. However, this attribute can be added for other agents (e.g. Greta). The spatial orientation attribute is illustrated as in Figure 7.

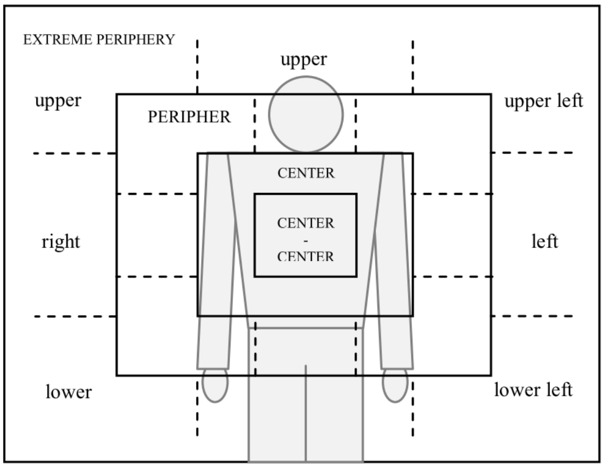


Fig. 6. Symbolic gesture space [9].

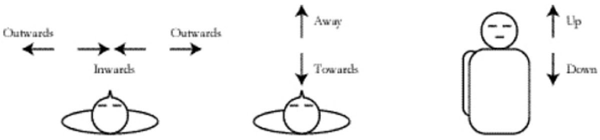


Fig. 7. Spatial orientation specification [11].

Movement Space Specification Each symbolic position is translated into concrete values of a fixed set of robot joints when the gestures are realized. In our case, they are four NAO joints: ElbowRoll, ElbowYaw, ShoulderPitch and ShoulderRoll. In order to overcome the limited gesture movement space of the robot, we have to predefine a finite set of wrist positions possible for the robot as shown in Table 8. In addition to the set of 105 possible wrist positions (i.e. following the gesture space of McNeill), two wrist positions are added to specify relax positions. These positions are used in the retraction phase of gesture. The first position indicates a full relax position (i.e. two hands are let loose along the body) while the second one indicates a partial relax position (i.e. one or two hands are retracted partially). Depending on the available time allocated to the retraction phase, one relax position is selected and used by the system.

Code	ArmX	ArmY	ArmZ	Joint values(LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll)
000	XEP	YUpperEP	ZNear	(-96.156,42.3614,49.9201,-1.84332)
001	XEP	YUpperEP	ZMiddle	(-77.0835,36.209,50.4474,-1.84332)
002	XEP	YUpperEP	ZFar	(-50.5401,35.9453,49.9201,-2.98591)
010	XEP	YUpperP	ZNear	(-97.3864,32.2539,30.3202,-7.20472)
...	...	...	...	...

Fig. 8. Key arm positions.

The other attributes such as palm orientation and hand shape are calculated automatically by the system when creating the animation from information indicated in the corresponding gesture prototype loaded from the lexicon. Due to physical limitations of the NAO robot, some combinations of parameters described at symbolic level cannot be realized. In such cases, the mapping between the symbolic description and NAO joints is realized by choosing the most similar available position if this does not change the signification of the elaborated gesture. Otherwise, this gesture must be deleted from the lexicon, or be replaced by another possible one which has a similar meaning.

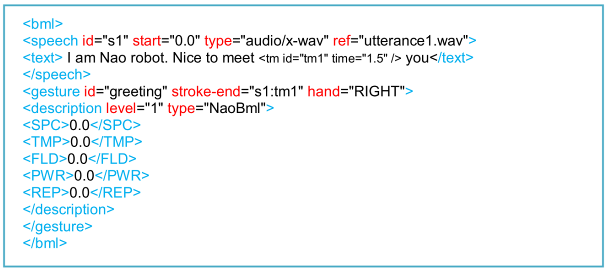
Position(from\to)	000	001	002	010	...
000	0	0.14	0.21	0.13	...
001	0.14	0	0.21	0.13	...
002	0.23	0.12	0	0.13	...
010	0.12	0.12	0.2	0	...
...	...	...	...	...	...

Fig. 9. Movement durations.

Movement Speed Specification Because the robot has limited movement speed, we need to have a procedure to verify the temporal feasibility of gesture actions. That means the system ought to estimate the minimal duration of a hand-arm movement from one position to another position in a gesture action as well as between two consecutive gestures. However, the Nao robot does not allow us to predict these durations before realizing real movements. Hence, we have to pre-estimate the necessary time between any two hand-arm positions in the gesture movement space, as shown in Table 9. The results in this table are used to calculate the duration of gesture phases in a gesture in order to eliminate inappropriate gestures (i.e. the allocated time is less than the necessary time to do the gesture) and to coordinate gestures with speech.

3.2 Gesture realization

The main task of this module is to create animation described in BML messages received from the Behavior Planner. In our system, a BML message contains information of gestures and speech to be realized. An example of BML message is shown in Figure 10.



```
<bml>
<speech id="s1" start="0.0" type="audio/x-wav" ref="utterance1.wav">
<text> I am Nao robot. Nice to meet <tm id="tm1" time="1.5" /> you</text>
</speech>
<gesture id="greeting" stroke-end="s1:tm1" hand="RIGHT">
<description level="1" type="NaoBml">
<SPC>0.0</SPC>
<TMP>0.0</TMP>
<FLD>0.0</FLD>
<PWR>0.0</PWR>
<REP>0.0</REP>
</description>
</gesture>
</bml>
```

Fig. 10. An example of BML message.

As outlined in Figure 3, the BML Parser module receives and analyses a BML message to initialize objects necessary to create the animation. Then it loads corresponding gestures' description from the gesture repository.

From the configuration and expected timing information of gestures indicated in the BML the Scheduling module calculates absolute time as well as the form of trajectory for each gesture while taking into account gesture expressivity parameters (e.g. the duration of gesture stroke phase is decreased when the temporal extension (TMP) is increased and vice-versa). At this stage, the system verifies the feasibility of gestures to eliminate inappropriate ones or cancel optional phases (i.e. preparation, retraction) of a certain gesture. If available time (provided by FML and computed by the speech synthesizer at the Behavior Planner) is not enough to do a gesture or this gesture is in conflict with the previous one (i.e. it starts while the previous has not yet finished the stroke phase), it is eliminated. In the case that a certain gesture starts before the ending of the retraction phase but after the stroke phase of the current phase, the retraction phase of the current phase is canceled.

In our system, we focus more on the synchronization of gestures with speech. This synchronization is realized by adapting the timing of the gestures to the speech's timing. It means the temporal information of gestures within *bml* tag are relative to the speech (cf. Figure 10). They are specified through time markers.

As shown in Figure 11, they are encoded by seven sync points: *start*, *ready*, *stroke-start*, *stroke*, *stroke-end*, *relax* and *end* [6]. These sync points divide a gesture action into certain phases such as preparation, stroke, retraction and hold phases as defined by Kendon [4]. The most meaningful part occurs between the stroke-start and the stroke-end (i.e. the stroke phase). According to McNeill's observations [9], a gesture always coincides or lightly precedes speech. In our system, the synchronization between gesture and speech is ensured by forcing the starting time of the stroke phase to coincide with the stressed syllables. The system has to pre-estimate the time required for realizing the preparation phase, in order to make sure that the stroke happens on the stressed syllables. This pre-estimation is done by calculating the distance between current hand-arm position and the next desired positions and by computing the time it takes to perform the trajectory. The results of this step are obtained by using values in the Tables 8 and 9.

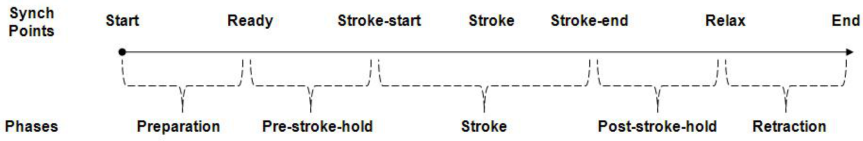


Fig. 11. Gesture phases and synchronization points .

The last Execution module (cf. Figure 3) translates gesture descriptions in the scripts into joint values of the robot. The symbolic position of the robot hand-arm (i.e. the combination of three values within BML tags respectively: *horizontal-location*, *vertical-location* and *location-distance*) is translated into concrete values of four robot joints: ElbowRoll, ElbowYaw, ShoulderPitch, ShoulderRoll using Table 8. The shape of the robot hands (i.e. the value indicated within *hand-shape* tag) is translated into the value of the robot joints, RHand and LHand respectively. The palm orientation (i.e. the value specified within *palm-orientation* tag) and the direction of extended wrist concern the wrist joints. As Nao has only the WristYaw joint, there is no symbolic description for the direction of the extended wrist in the gesture description. For the palm orientation, this value is translated into the robot joint WristYaw by calculating the current orientation and the desired orientation of the palm. Finally, the joint values and the timing of movements are sent to the robot. The animation is obtained by interpolating between joint values with the robot built-in proprietary procedures [2]. Data to be sent to the robot (i.e. timed joint values) are sent to a waiting list. This mechanism allows the system to receive and process a series of BML messages continuously. Certain BML messages can be executed with a higher priority order by using an attribute specifying its priority level. This can be used when the robot wants to suspend its current actions to do an exceptional gesture (e.g. do greeting gesture to a new listener while telling story).

4 Conclusion and future work

In this paper, we have presented an expressive gesture model for the humanoid robot NAO. The realization of the gestures are synchronized with speech. Intrinsic constraints (e.g. joint and speed limits) are also taken into account.

In the future, we plan first to improve the movement speed specification with the Fitt's Law (i.e. simulating human movement). Then the system needs to be equipped with a feedback mechanism. This mechanism is important to re-adapt the actual state of the robot while scheduling gestures. Last but not least, we aim to valide the model through perceptive evaluations. Accordingly, we will test how expressive the robot is perceived when reading a story.

5 Acknowledgment

This work has been funded by the French ANR project GVLEX (http://www.gvlex.com).

References

1. Calbris, G.: Contribution à une analyse sémiologique de la mimique faciale et gestuelle française dans ses rapports avec la communication verbale. Ph.D. thesis (1983)
2. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: Mechatronic design of NAO humanoid. In: Robotics and Automation. ICRA'09. pp. 769–774. IEEE (2009)
3. Hartmann, B., Mancini, M., Pelachaud, C.: Implementing expressive gesture synthesis for embodied conversational agents. gesture in human-Computer Interaction and Simulation pp. 188–199 (2006)
4. Kendon, A.: Gesture: Visible action as utterance. Cambridge University Press (2004)
5. Kipp, M., Neff, M., Albrecht, I.: An annotation scheme for conversational gestures: How to economically capture timing and form. Language Resources and Evaluation 41(3), 325–339 (2007)
6. Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Thórisson, K., Vilhjálmsson, H.: Towards a common framework for multimodal generation: The behavior markup language. In: Intelligent Virtual Agents. pp. 205–217. Springer (2006)
7. Kushida, K., Nishimura, Y., Dohi, H., Ishizuka, M., Takeuchi, J., Tsujino, H.: Humanoid robot presentation through multimodal presentation markup language mpml-hr. In: AAMAS-05 Workshop on Creating Bonds with Humanoids. IEEE (2005)
8. Martin, J.C.: The contact video corpus (2009)
9. McNeill, D.: Hand and mind: What gestures reveal about thought. University of Chicago Press (1992)
10. Ng-Thow-Hing, V., Luo, P., Okita, S.: Synchronized gesture and speech production for humanoid robots. In: Intelligent Robots and Systems (IROS2010). pp. 4617–4624. IEEE

11. Pelachaud, C.: Modelling multimodal expression of emotion in a virtual agent. Philosophical Transactions of the Royal Society B: Biological Sciences 364(1535), 3539 (2009)
12. Pot, E., Monceaux, J., Gelin, R., Maisonnier, B.: Choregraphe: a graphical tool for humanoid robot programming. In: Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on. pp. 46–51. IEEE (2009)
13. Prillwitz, S.: HamNoSys Version 2.0: Hamburg notation system for sign languages: An introductory guide. Signum (1989)
14. Rich, C., Ponsleur, B., Holroyd, A., Sidner, C.: Recognizing engagement in human-robot interaction. In: Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction. pp. 375–382. ACM (2010)
15. Salem, M., Kopp, S., Wachsmuth, I., Joublin, F.: Generating robot gesture using a virtual agent framework. In: Intelligent Robots and Systems (IROS2010). pp. 3592–3597. IEEE