Generating Co-speech Gestures for the Humanoid Robot NAO through BML

Quoc Anh Le and Catherine Pelachaud

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

Abstract. We extend and develop an existing virtual agent system to generate communicative gestures for different embodiments (i.e. virtual or physical agents). This paper presents our ongoing work on an implementation of this system for the NAO humanoid robot. From a specification of multi-modal behaviors encoded with the behavior markup language, BML, the system synchronizes and realizes the verbal and nonverbal behaviors on the robot.

Keywords: Conversational humanoid robot, expressive gestures, gesture-speech production and synchronization, Human-Robot Interaction, NAO, GRETA, FML, BML, SAIBA.

1 Introduction

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

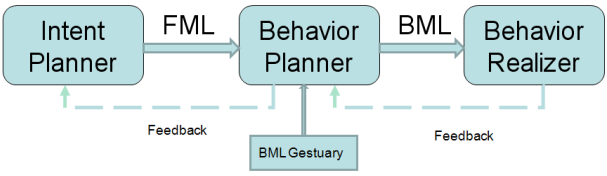

Fig. 1. The SAIBA framework for generating multimodal behavior

The results of the first module is the input for the second module presented through an interface described with a representation markup language, named

E. Efthimiou, G. Kouroupetroglou, S.-E. Fotinea (Eds.): GW 2011, LNAI 7206, pp. 228–237, 2012.

FML (i.e Function Markup Language). The output of the second module is encoded with another representation language, named BML (i.e. Behavior Markup Language) [6] and then sent to the third module. Both FML and BML are XML-based languages and they do not refer to any particular specification agent (e.g. its wrist joint).

From given communicative intentions, the system selects and plans gestures taken from a repository of gestures , called Gestural Lexicon or Gestuary (cf. Figure 1). In the repository, gestures are described symbolically with an extension of the BML representation language. Then the system calculates the timing of the selected gestures to be synchronized with speech. After that, the gestures are instantiated as robot joint values and sent to the robot in order to execute the hand-arm movements.

Our aim is 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 movement capacities (e.g. the robot can move its legs and torso but does not have facial expression and has very limited hand-arm movements compared to the agent). Therefore, the nonverbal behaviors to be displayed by the robot may be different from those of the virtual agent. For instance, the 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 the robot and the virtual agent, while taking into account their physical constraint, we consider two repertoires of gestures, one for the robot and another 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 both agents and unchanged from the GRETA system. From the BML scripts outputted by the Behavior Planner, we instantiate BML tags from either gesture repertoires. That is, given a set of intentions and emotions to convey, the GRETA system, through the Behavior Planning, the corresponding sequence of behaviors specified with BML. The Behavior Realizer module has been developed to create the animation for both agents with different behavior capabilities. Figure 2 presents an overview of our system.

In this paper, we presents our current implementation of the proposed expressive gesture model for the NAO humanoid robot. This work is conducted within the framework of the French Nation Agency for Research project, named GVLEX (Gesture and Voice for an Expressive Lecture), whose objective is to build an expressive robot able to display communicative gestures with different behavior qualities while telling a story. Whistle other partners of the project deal

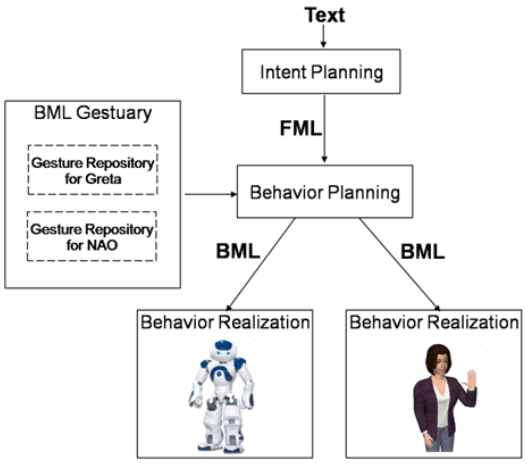

Fig. 2. An overview of the proposed system

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 based on 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 GRETA agent [3].

The paper is structured as follows. The next section, state of the art describes some recent initiatives in controlling humanoid robot hand-arm gestures. Then, Section 3 presents in detail the design and implementation of a gesture database and a behavior realizer for the robot. Section 4 concludes and proposes some future works.

2 State of the Art

Several initiatives have been proposed recently to control multi-modal behaviors of a humanoid robot. Salem et al. [14] use the gesture engine of the MAX virtual agent to drive the ASIMO humanoid robot. Rich et al. [13] implement a system following an event-driven architecture to solve the problem of unpredictability in performance of their MELVIN humanoid robot. Meanwhile, Ng-Thow-Hing et al. [10] develop a system that takes any text as input to select and produce the corresponding gestures for the ASIMO robot. In this system Ng-Thow-Hing added some parameters for expressivity to make gestures more expressive such as tension, smooth and timing of gesture trajectories. These parameters correspond to the power, fluidity and temporal extend in our system. In [7] Kushida et al. 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 MURML [14], BML [13], MPML-HR [7], etc. The synchronization of gestures with speech is guaranteed by adapting the gesture movements to the timing of the speech [14,10]. This is also the method 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 [14], or to improve the synchronization of gestures with speech [13].

Our system has some differences from these works. It focuses not only on the coordination of gestures and speech but also on the signification and the expressivity of gestures performed by the robot. In fact, the gesture signification is studied carefully when elaborating a repertoire of robot gestures. In terms of gesture expressivity, it is enhanced by adding a set of gesture dimension parameters such as spatial extension (SPC), temporal extension (TMP).

3 System Design and Implementation

The proposed model is developed based on the GRETA system. It uses its existing Behavior Planner module to select and plan multi-modal behaviors. A new Behavior Realizer module has been developed to adapt to the behavior capabilities of the robot. The main objective of this module is to generate the animations, which will be displayed by the robot from received BML messages. 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 realize selected and planned gestures on the robot. Figure 3 shows different steps of the system.

3.1 Gesture Database

Gestures are described symbolically using a BML-based extension language and are stored in a lexicon. All entries of the lexicon are tested to guarantee their realizability on the robot (e.g. avoid collision or conflict between robot joints when doing a gesture, or avoid singular positions where the robot hands cannot reach). The gestures are instantiated dynamically into joint values of the robot when creating the animation in real-time. The instantiation values according to the values of their expressivity parameters.

Gesture Annotations. The elaboration of symbolic gestures in the robot lexicon is based on gesture annotations extracted from a Storytelling Video Corpus, which was recorded and annotated by Jean-Claude Martin et al. [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 such as its category (i.e. iconic, beat, metaphoric and deictic), its

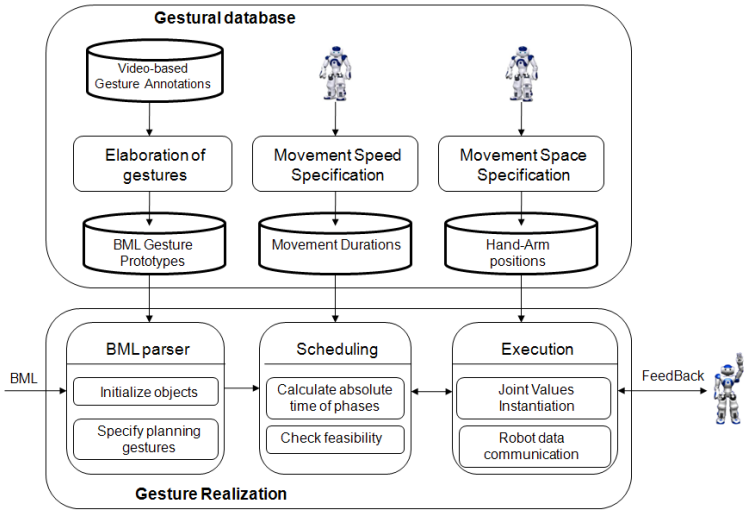

Fig. 3. Steps in the system

duration and which hand is being used, etc (cf. Figure 4). Based on the shape of gestures captured from the video and their annotated information, we have elaborated a corresponding symbolic gesture repository.

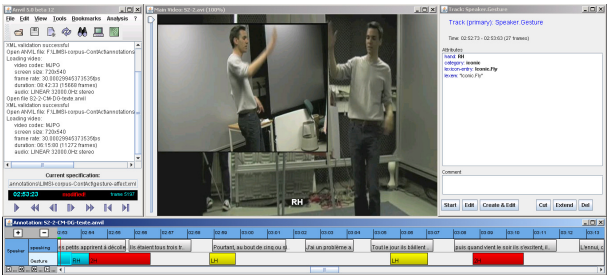

Fig. 4. Gestural annotations from a video corpus with the Anvil tool

Gesture Specification. We have proposed a new XML schema as an extension of the BML language to describe symbolically gestures in the repository (i.e. lexicon). The specification of a gesture relies on the gestural description of McNeill [9], the gesture hierarchy of Kendon [4] and some notions from the HamNoSys system [12]. As a result, a hand gesture action may be divided into several phases of wrist movements. The stroke phase carries the meaning of the gesture. It 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

to the relax positions or positions initialized for the next gesture (cf. Figure 7). In the lexicon, only the description of the 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 namely *vertical location* that corresponds to the Y axis, *horizontal location* that corresponds to the X axis, and *location distance* 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 the 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). 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 (here, above the head), the hand shape (open) and the palm orientation (forward). The two key poses are different from only one symbolic value on the 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. the GRETA agent).

Movement Space Specification. Each symbolic position is translated into concrete joint values of the robot joints when the gestures are realized. In our case, these include four NAO joints: *ElbowRoll*, *ElbowYaw*, *ShoulderPitch* and *ShoulderRoll*. In addition to the set of 105 possible wrist positions (i.e. following the gesture space of McNeill, see Table 1), two wrist positions are added to specify relax positions. These positions are used in the retraction phase of a 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.

Table 1. Specification of key-arm positions

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)
...	...	...	...	...

Table 2. Specification of gesture movement durations

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	...
...	...	...	...	...	...

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 that moves robot wrist from one position to another one 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 minimal time between any two hand-arm positions in the gesture movement space, as shown in Table 2. The results in this table are used to calculate the duration of gesture phases to eliminate inappropriate gestures (i.e. the allocated time is less than the necessary time to perform the gesture) and to coordinate gestures with speech.

3.2 Gesture Realization

The main task of this module is to compute the animation described in BML messages received from the Behavior Planner. An example of the format of a BML message is shown in Figure 6.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE bml SYSTEM "bml.dtd" []>
<bml>
<tm id='tm1'/>
what are you
<tm id='tm2'/>
doing
<tm id='tm3'/>
here
</speech>
    <gesture id="performative"  start='s1:tm2' end='s1:tm3'>
        <description level="1" type="gretabml">
            <reference>adjectival=WARNING_HAND</reference>
            <intensity>1.00</intensity>
            <stroke time="2.462"/>
            <FLD.max>1.00</FLD.max>
            <FLD.min>-1.00</FLD.min>
            <FLD.value>-0.85</FLD.value>
            <OAC.max>1.00</OAC.max>
            <OAC.min>0.00</OAC.min>
            <OAC.value>0.75</OAC.value>
            <PWR.max>1.00</PWR.max>
            <PWR.min>-1.00</PWR.min>
            <PWR.value>0.90</PWR.value>
            <REP.max>1.00</REP.max>
            <REP.min>-1.00</REP.min>
            <REP.value>-0.10</REP.value>
            <SPC.max>1.00</SPC.max>
            <SPC.min>-1.00</SPC.min>
            <SPC.value>0.50</SPC.value>
            <TMP.max>1.00</TMP.max>
            <TMP.min>-1.00</TMP.min>
            <TMP.value>0.55</TMP.value>
```

Fig. 6. An example of the BML description

In our system, we focus on the synchronization of gestures with speech. This synchronization is realized by adapting the timing of the gestures to the speech timing. It means the temporal information of gestures within BML tags are relative to the speech (cf. Figure 6). They are specified through time markers. As shown in Figure 7, 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 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 of the speech. This pre-estimation is done by calculating the distance between current hand-arm position and the next desired positions. This is also calculated by computing the time it takes to perform the trajectory. The results of this step are obtained by using values in the Tables 1 and 2.

The last Execution module (cf. Figure 3) translates gesture descriptions into joint values of the robot. The symbolic positions of the robot hand-arm (i.e. the combination of three values within BML tags respectively: *horizontal-location*, *vertical-location* and *location-distance*) are translated into concrete values of four robot joints: *ElbowRoll*, *ElbowYaw*, *ShoulderPitch*, *ShoulderRoll* using Table 1. 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 concerns 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

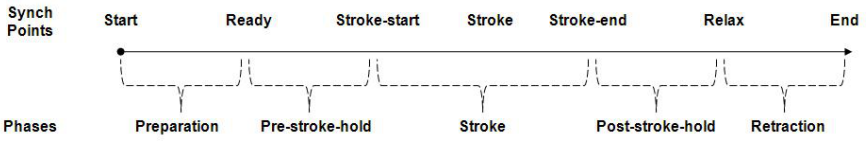

Fig. 7. Gesture phases and synchronization points

into a robot joint, namely *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. make a 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.

Gesture expressivity is calculated in real-time by setting values to a set of parameters that modulate gestural animation. In the near future, we aim at improving the movement speed specification with the Fitt's Law (i.e. simulating human movement). So far, the model has been developed for arm-hand gestures only. In the next stage, we will extend the system for head and torso gestures. Then, the system needs to be equipped with a feedback mechanism, which is important to re-adapt the actual state of the robot while scheduling gestures. Last but not least, we aim to validate our model through perceptive evaluations. We will test how expressive the robot is perceived when reading a story.

Acknowledgment. This work has been partially funded by the French ANR GVLEX project.

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 2009, pp. 769–774. IEEE Press (2009)

3. Hartmann, B., Mancini, M., Pelachaud, C.: Implementing Expressive Gesture Synthesis for Embodied Conversational Agents. In: Gibet, S., Courty, N., Kamp, J.-F. (eds.) GW 2005. LNCS (LNAI), vol. 3881, pp. 188–199. Springer, Heidelberg (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.N., Pelachaud, C., Pirker, H., Thórisson, K.R., Vilhjálmsson, H.: Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 205–217. Springer, Heidelberg (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 2005 Workshop on Creating Bonds with Humanoids. IEEE Press (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 (IROS), pp. 4617–4624. IEEE Press (2010)
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. Prillwitz, S., Leven, R., Zienert, H., Hanke, T., Henning, J., et al.: HamNoSys Version 2.0: Hamburg notation system for sign languages: An Introductory Guide, vol. 5. University of Hamburg (1989)
13. 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 Press (2010)
14. Salem, M., Kopp, S., Wachsmuth, I., Joublin, F.: Generating robot gesture using a virtual agent framework. In: Intelligent Robots and Systems (IROS), pp. 3592–3597. IEEE Press (2010)