Design and implementation of an expressive gesture model for a humanoid robot

Quoc Anh Le, Souheïl Hanoune, Catherine Pelachaud

▶ **To cite this version:**

Quoc Anh Le, Souheïl Hanoune, Catherine Pelachaud. Design and implementation of an expressive gesture model for a humanoid robot. Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference, Oct 2011, Bled, Slovenia. pp.134 - 140, ⟨10.1109/Humanoids.2011.6100857⟩. ⟨hal-00730800⟩

HAL Id: hal-00730800

https://hal.science/hal-00730800v1

Submitted on 11 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Design and implementation of an expressive gesture model for a humanoid robot

Quoc Anh Le
Telecom ParisTech, France
quoc-anh.le@telecom-paristech.fr

Souheil Hanoune
Telecom ParisTech, France
hanoune@telecom-paristech.fr

Catherine Pelachaud
CNRS, LTCI Telecom ParisTech, France
catherine.pelachaud@telecom-paristech.fr

***Abstract*— We aim at equipping the humanoid robot NAO with the capacity of performing expressive communicative gestures while telling a story. Given a set of intentions and emotions to convey, our system selects the corresponding gestures from a gestural database, called lexicon. Then it calculates the gestures to be expressive and plans their timing 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. The robot has certain physical constraints to be addressed such as the limits of movement space and joint speed. This article presents our ongoing work on a gesture model generating co-verbal gestures for the robot while taking into account these constraints.**

Keywords-humanoid; gesture; expressivity; lexicon; BML; SAIBA; GRETA; NAO

I. INTRODUCTION

Many studies have shown the importance of expressive gestures in communicating messages as well as in expressing emotions. They are necessary for the speaker to formulate his thoughts [10]. They can convey complementary, supplementary or even contradictory information to the one indicated by speech [11].

In the domain of humanoid robots, the communication of emotion through gestures have experimented and obtained results [22, 24]. The objective of our work is to equip a physical humanoid robot with the capability of producing such expressive gestures while talking. This research is conducted within the frame of the French ANR project GVLEX that has started since 2009 and lasts for 3 years. The project aims to model the humanoid robot, NAO [8], developed by Aldebaran, to read a story in an expressive manner to children for several minutes without boring them. While other partners of the GVLEX project deal with expressive voice, our work focuses on expressive behaviors, especially on gestures [17].

To reach this objective, we have extended and developed our existing virtual agent platform GRETA [1] to be adapted to the robot. Using a virtual agent framework for a physical robot raises several issues to be addressed because the robot has the limit of movement space and joint speed. The idea is to use the same representation language to control both virtual and physical agents [18]. This allows using the same algorithms for selecting and planning gestures but different algorithms for creating the animation. The work presented in this paper concerns mainly the animation of the humanoid robot, in which displayed gestures are ensured to be tightly tied to speech.

In detail, the GRETA system calculates the nonverbal behaviors that the robot must show to communicate a text in a certain way. The selection and planning of the gestures are based on the information that enriched the input text. Once selected, the gestures are planned to be expressive and to be synchronized with speech, then they are realized by the robot. To calculate their animation, the gestures are transformed into key poses. Each key pose contains the joint values of the robot and the timing of its movement. The animation module is script-based. That means the animation is specified and described with the multimodal representation language BML [3]. As the robot has some physical constraints, the scripts are instantiated so as to be feasible for the robot.

The gestures of the robot are stored in a library of behaviors, called Lexicon, and described symbolically with an extension of the language BML. These gestures are elaborated using gestural annotations extracted from a storytelling video corpus [4]. Each gesture in the robot lexicon should be executable by the robot (e.g. avoid collisions or singular positions where the robot hand cannot reach). When gestures are realized, their expressivity is increased by considering parameters of the gestural dimensions. We have designed and implemented a set of gestural dimensions such as the amplitude (SPC), fluidity (FLD), power (PWR) and the speed of gestures (TMP) for the virtual agent Greta [2]. The objective is to realizing such a model for the robot.

This paper is structured as follows. The next section presents some recent initiatives in generating humanoid robot gestures. Then, Section 3 shows an overview of our system to be implemented. Section 4 gives some observations obtained from gesture generation experiments for the Nao robot. In Section 5, we talk about a method for building a gestural database overcoming physical constraints of the robot. Section 6 shows how robot gestures with expressivity are produced and realized. Section 7 concludes the paper and proposes some future works.

II. STATE OF THE ART

There are some existing approaches to create gestural animation for humanoid robots. One way is to elaborate a library of gestures as a set of predefined animation scripts with

fixed hand-arm movements [21, 23]. Another way is to calculate the trajectory of gestures on the fly [7,9,25]. Our method follows the second approach that allows us to adjust gestures online with expressivities for a certain intention. This section presents several systems which have been developed recently to generate gestures in realtime for humanoid robots. The robot gestures accompanying speech are created in systems described in [7, 9,15]. Salem et al. [7] and Ng-Thow-Hing et al. [9] produce co-verbal gestures to be performed by the robot ASIMO. Similarly to the system of Kim et al. [15], the system of Ng-Thow-Hing is geared toward the selection of gestures corresponding to a given text, while the system of Salem concentrates on improving gestural trajectories. All of them have a mechanism for synchronizing gestures and speech. However, only the system of Salem has a cross-modal adaptation mechanism which not only adapts gestures to speech but also adjusts the timing of running speech to satisfy the duration of gestural movements. Other systems as presented in [14, 17], which do not deal with the synchronization of gestures and speech, use different approaches to generate robot gestures. Rai et al. [14] present an architecture based on a knowledge-based system (KBS) to generate intelligent gestures. Using the rule engine Jess (Java Expert System Shell) as KBS, they implemented the specific facts of the robot as modules (e.g. Hand_Close, Arm_Positioning, etc), so that their rule-based system can generate different gestures without ambiguities. Hiraiwa et al. [16] use EMG signals extracted from a human to drive a robot arm and hand gesture. The robot replicates the gestures of the human in a quite precise manner.

There are some differences between our system and the others. Our system follows the SAIBA framework [3], a standard architecture for multimodal behavior generation. Its gesture lexicon is considered an external parameter that can be modified to be adapted to a specific robot (i.e. a robot with physical constraints). Additionally, in our system, gestural expressivities are taken into account when creating gesture animations for the robot.

III. SYSTEM OVERVIEW

Our proposed approach relies on the system of the conversational agent Greta [1] following the architecture of SAIBA [3] (cf. Figure 1). It consists of three separated modules: (i) the first module, Intent Planning, defines the communicative intents to be conveyed; (ii) the second, Behavior Planning, plans the corresponding multimodal behaviors to be realized; (iii) and the third module, Behavior Realizer, synchronizes and realizes the planned behaviors. The results of the first module is the input of the second module through an interface de- scribed with a representation markup language, named FML (Function Markup Language). The output of the second module is encoded with another representation language, named BML [3] and then sent to the third module. Both languages FML and BML are XML-based and do not refer to specific animation parameters of agents (e.g. wrist joint).

We aim 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 motion capacities (e.g. the robot can move its legs and torso but does not have facial expression and has very limited hand-arm movements). Therefore the nonverbal behaviors to be displayed by the robot should be different from those of the virtual agent. For instance, the three fingers of the robot hand can only open or close together; 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 the communicative behaviors of the robot and of the virtual agent, while taking into account the physical constraint of both, two lexicons have been built, one for the robot and one for the agent [18]. The Behavior Planning module of the GRETA framework remains the same. From a BML message outputted by the Behavior Planner, we instantiate the BML tags from either gestural 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.

Figure 1. The SAIBA architecture

At the Behavior Realizer layer, some extensions are added to generate the animation specific to different embodiments (i.e. Nao and Greta). Firstly, the BML message received from Behavior Planner is interpreted and scheduled by a sub-layer called Animations Computation. This module is common to both agents. Then, an embodiment dependent sub-layer, namely Animation Production, generates and executes the animation corresponding to the specific implementation of agent. Figure 2 presents an overview of our system.

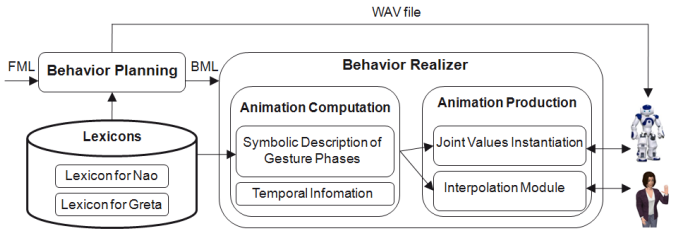

Figure 2. System Overview

To ensure that both the robot and the virtual agent convey similar information, their gestural repertoires have entries for the same list of communicative intentions. The elaboration of repertoires encompasses the notion of gestural family with variants [12]. Gestures from the same family convey similar meanings but may differ in their shape (i.e. the element deictic exists in both lexicons; it corresponds to an extended finger or to an arm extension).

IV. EXPERIMENTAL OBSERVATION

We have tested on the gesture production for the robot and the virtual agent. The objective is to show that they have differences when doing gestures. The test is based on a short

BML script of the French story "Three little pieces of night", in which both agents used the same lexicon to make gestural movements. That means they do the same gestures and use the same gestural timing described in the scripts. From this test, some observations are drawn [20].

The robot starts the gestures earlier than the agent does but the stroke phase of the gestures arrive at the same time (see Figure 3). However, in most cases, the gesture movements are lagging behind compared to the uttered speech.

Figure 3. Differences in movement timing between the agents

Moreover, some gestures for the robot are eliminated because the allocated time is not enough to do them. It is due to the speed limit of the physical robot. Otherwise, jerky movements happen when the speed of movement is too fast.

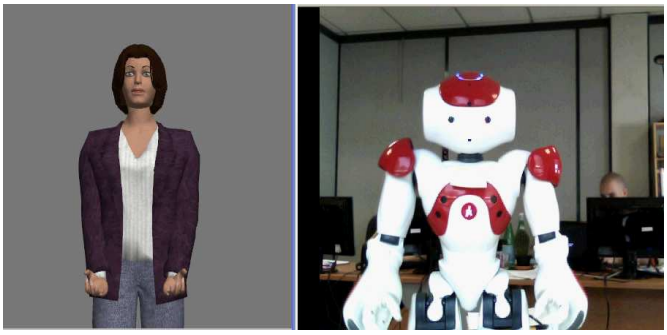

Figure 4. The gesture done by Greta is impossible for the robot

The test also showed that some gestures were unrealizable for the robot. Figure 4 illustrates one gestural configuration: the palm orientation is up and the extended wrist orientation is forwards while the hand position is near at the relax position. This gesture cannot be realized by the robot. The reason is that the robot does not have a dynamic wrist joints (i.e. wrist roll joint).

These observations showed the limit in movement space and in speed of the robot when it uses the same lexicon of the virtual agent. Thus, we decided to create a specific lexicon for the robot that takes into account the robot's limitations.

V. GESTURAL DATABASE

A. *Gesture Specification*

We have proposed a new XML notation to symbolically describe gestures in gestural repositories (i.e. lexicons). The specification of a gesture relies on the gestural description of McNeill [5], the gestural hierarchy of Kendon [6] and some notions from the HamNoSys system [13]. As a result, a gestural action may be divided into several phases of wrist movements, in which the obligatory phase is called stroke which carries 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 by the next gesture (cf. Figure 8).

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 <*vertical_location*> that corresponds to the Y axis, <*horizontal_location*> that corresponds to the X axis, and <*location_distance*> corresponding to the Z axis (e.g. distance of the hand with respect to the body) in a limited movement space [2].

```
<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>
```

Figure 5. An example of gesture specification

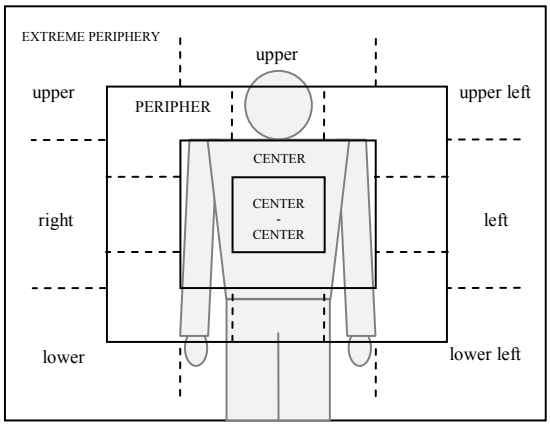

Figure 6. Symbolic gestural space.[5]

Following the gestural space proposed by McNeill [5], 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). They 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 gestural specification for the robot. However, this attribute can be added for other agents (e.g. Greta).

B. Predefined wrist positions and movement durations

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 I. In addition to the set of 105 possible wrist positions (i.e. following the gestural 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) and 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 I. 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)
…	…	…	…	…

The other attributes such as palm orientation and hand shape are calculated automatically by the system at the Animation Computation module.

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.

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 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 for a hand-arm movement between any two positions in the gesture movement space, as shown in Table II. For each couple of positions, we have two values (min:fitt). The first value corresponds to the minimal duration in which the robot can do the movement (ie. using maximal speed). The second value indicates a normal duration in which the robot make the movement with a human speed. The Fitt's Law is used to calculate these normal durations. The robot should use normal speed instead of the maximal speed excepting the case that normal duration is smaller than minimal duration (eg. the movement between two positions 002 and 010 in the table).

TABLE II. MOVEMENT DURATIONS

Position (from\to)	000	001	002	010	…
000	0	0.15:0.18388	0.25:0.28679	0.166:0.2270	…
001	0.15:0.18388	0	0.19:0.19552	0.147:0.2754	…
002	0.25:0.28679	0.19:0.19552	0	1.621:0.3501	…
010	0.166:0.2270	0.147:0.2754	1.621:0.3501	0	…
…	…	…	…	…	…

The results in this table are used to calculate the duration of gestural 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 schedule gestures with speech.

C. Gesture Elaboration

The elaboration of symbolic gestures in a lexicon is based on gestural annotations extracted from a Storytelling Video Corpus using the defined gesture specification. The video corpus was recorded and annotated by Jean-Claude Martin [4], a partner of the GVLEX project. To do 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 [19] is used to annotate gestural information. 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.

All gestural lexicon are tested to guarantee its realizability on the robot (e.g. avoid collision or conflict between robot joints when doing a gesture) using predefined values in Table I.

VI. GESTURE REALIZER

The main task of this module is to create animations 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 7.

In this example, the *speech* tag indicates the name of audio file as well as the start time to play the file by the robot. This file is created by a speech synthesis module in the Behavior Planner. The time marker (e.g. *tm1*) is used to synchronize with the *greeting* gesture. The timing of the gesture is relative to the speech through the time maker. In the *gesture* tag, the unique identification *id* is used to refer to a symbolic gestural description in the lexicon. In this case, it refers to the gestural description in Figure 5. The values of expressivity parameters are specified within the *SPC, TMP, FLD, PWR, REP* tags respectively. They have a value from -1 to 1, in which 0 corresponds to a neutral state of gesture (i.e. normal movements).

```
<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="s1:tm1" hand="RIGHT" REP="1">
<description level="1" type="NaoBml">
<SPC>1.0</SPC>
<TMP>0.5</TMP>
<FLD>0.0</FLD>
<PWR>0.0</PWR>
<STF>1.0</STF>
</description>
</gesture>
</bml>
```

Figure 7. An example of BML message

The Gesture Realizer is divided into two main stages: the first one, called Animation Computation interprets the received BML message and schedules the gestures. The second one, Animation Production generates and executes gestural animation. In the following subsections we present these modules in details.

A. Animation Computation

At this stage, the gestures described in the BML message are initialized. The system loads the symbolical description of the corresponding gestures in the robot lexicon (e.g. the gesture in Figure 7 whose *id* tag is "*greeting*" is instantiated with the symbolic description presented in Figure 5). After that, it verifies the feasibility of the gestures to eliminate inappropriate ones. Then it schedules the gestures to be realized. The expressivity parameters of each gesture are taken into account when the timing and configuration of the gesture are calculated.

1) *Synchronization of gestures with speech:* In our system, the synchronization between gestural signals and speech 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 7). They are specified through time markers. As shown in Figure 8, they are encoded by seven sync points: *start*, *ready*, *stroke-start*, *stroke*, *stroke-end*, *relax* and *end* [3]. These sync points divide a gestural action into certain phases such as preparation, stroke, retraction and hold phases as defined by Kendon [6]. The most meaningful part occurs between the *stroke-start* and the *stroke-end* (i.e. the stroke phase).

According to McNeill's observations [5], 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 I and II. In case the allocated time is not enough to perform the preparation phase (i.e. the allocated time is less than the minimal duration to do a movement from *start* to *ready* position), the priority level of the current gesture in the sentence is taken into consideration in order to decide whether this gesture can be canceled and leaving free time to prepare for the next gesture. If this is the most significant one, the previous gesture, if exists, should be skipped so that we have more time to perform the next one which is more important. Otherwise,, the timing of *start* and *ready* sync points are specified. The *ready* point coincides the *stroke-start* point unless a pre-hold-phase exists (e.g. *start*=0.6 s and *ready*=*stroke-start*=1.25 s for the example in Figure 7). If the allocated time is too long, a hold phase is added. A threshold is calculated by using the Fitts's law (i.e. simulating human movement) to determine the normal speed of human movements (see Table II). The retraction phase is optional. This depends on the available time in the retraction phase. If the available time is not enough to move hands to the partial relax position, the retraction phase will be canceled. Otherwise, the timing of *relax* and *end* sync points is calculated. The *relax* point coincides with the *stroke-end* point if there is no post-hold-phase (e.g. *relax*=*stroke-end*=1.5 s and *end*=2.15 for the example in Figure 7).

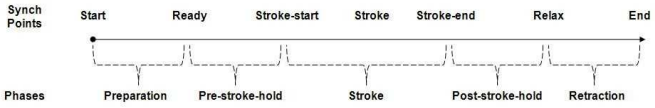

Figure 8. Gestural phases and synchronization points

2) *Expressivity Parameters:* The system applies the expressivity parameters while it is planning the gestures. So far, we have implemented three among the available expressivity parameters [2]. They are temporal extent, spacial extent and stroke repetition. Temporal Extent (TMP) modifies the duration of stroke phase. If the TMP value increases, the duration of gesture is decreased. That means the speed of the movements execution is faster. However, the time of *stroke-end* point is not changed so that the synchronization between gesture and speech is maintained. Consequentially, the *start* and *stroke-start* are later. Concerning Spatial Extent (SPC), it modifies the amplitude of movement. If the SPC value increases, the three values (vertical, horizontal and distance) of the wrist position are increased. The repetition (REP) calculates the number of repetiting stroke phase in a gesture action. The exact timing of each stroke repetition is specified

in the BML message. The duration of the complete gesture increases linearly with the REP value. In addition to existing expressivity parameters, we have also implemented a new parameter, namely Stiffness (STF) that effects the force of gestural movements. The remaining parameters, Fluidity (FLD) and Power (PWR) are planned to realize in the next work. A combination of these parameters defines an emotional state of the robot. For instance for the sadness state, gestural movement should be slower and narrower.

The result of the Animation Computation is a set of animation scripts. Each script contains the symbolic description and timing of one gestural phase. We use the same the XML notation introduced in Section V to describe the gestural actions. The symbolic representation allows us to use the same algorithms for different agents (i.e. the Animation Computation is common for both agents, Greta and Nao). Figure 9 shows an example, in which three gestural phases (preparation, stroke and retraction) are planned to be executed. The timing of each phase is based on the sync points (see Figure 8). For each phase, only the description of the destination position is given. For instance, the preparation moves the right hand from the current position (i.e. relax position) to the stroke-start position, only the description of gesture at the stroke-start's moment is specified.

```
<animations>
    <animation phase="preparation" start="0.6" end="1.25">
            <!-- go to the stroke-start position- -->
            <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>
    </animation>
    <animation phase="stroke" start="1.25" end="1.5">
            <!-- go to the stroke-end position-->
            <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>
    </animation>
    <animation phase="retraction" start="1.5" end="2.15">
            <!-- go to the full relax position -->
    </animation>
</animations>
```

Figure 9. An example of animation scripts

B. *Animation Production*

To generate the animation parameters (i.e. joint values) from the given scripts, we use a Joint Values Instantiation module (see Figure 2). This module is specific to the Nao robot. It translates gestural descriptions in the scripts into joint values of the robot. First, 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 I. The shape of the robot hands (i.e. the value indicated within *hand-shape* tag) translated into the value of the robot joints, RHand and LHand respectively. This variable has a value from 0 to 1, in which 0 corresponds to close hand and 1 corresponds to open hand. 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 are not any symbolic descriptions for the direction of the extended wrist in the gestural 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 movement are sent to the robot. The animation is obtained by interpolating between joint values with robot built-in proprietary procedures [8].

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 comer while telling story).

VII. CONCLUSION AND FUTURE WORK

We have designed and implemented an expressive gesture model for the humanoid robot Nao. A gestural database overcoming certain physical constraints of the robot has been defined. From the set of key positions, the system can be extended to perform a combination of gestures regardless physical constraints. The model has a gestural lexicon as an external parameter that can be customized to be applied to similar humanoid robots. In this lexicon, only stroke phase of gestures are shaped, other phases and their schedule are calculated online by the system.

In the future, we plan first to complete the model with full expressivities. Then, the system needs to be equipped with a feedback mechanism. This mechanism is important. It ensures the system to select and plan next gestural actions correctly taking into account the actual states of the robot. For instance the robot should stop gesturing if it falls down. Finally we aim to validate the 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 project GVLEX (http://www.gvlex.com).

REFERENCES

[1] C. Pelachaud, "Multimodal Expressive Embodied Conversational Agents" in Proceedings of the 13th annual ACM international conference on Multimedia, pp. 683–689, 2005.

[2] B. Hartmann, M. Mancini, C. Pelachaud, "Implementing expressive gesture synthesis for embodied conversational agents" in International on Gesture Workshop, France, May 2005, pp.188–199.

[3] S. Kopp, B. Krenn, S. Marsella, A. Marshall, C. Pelachaud, H. Pirker, K. Thorisson, H. Vilhjalmsson, "Towards a common framework for multimodal generation: The behavior markup language" in Intelligent Virtual Agents, pp. 205-217, IVA 2006.

[4] J.C. Martin, "The contact video corpus", France, 2009.

[5] D. McNeill, "Hand and mind: What gestures reveal about thought" in University of Chicago Press, 1992.

[6] A. Kendon, "Gesture: Visible action as utterance" in Cambridge University Press, 2004.

[7] M. Salem, S. Kopp, I. Wachsmuth, F. Joublin, "Towards an Integrated Model of Speech and Gesture Production for Multi-Modal Robot Behavior" in Proceedings of the International Symposium on Robot and Human Interactive Communication, ROMAN 2010.

[8] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, B. Maisonnier, "Mechatronic design of NAO humanoid" in Robotics and Automation" in International Conference on Robotics and Automation, pp. 769-774, ICRA 2009.

[9] V. Ng-Thow-Hing, P. Luo, S. Okita, "Synchronized Gesture and Speech Production for Humanoid Robots" in International Conference on Intelligent Robots and Systems, Taiwan, pp. 4617-4624, IROS 2010.

[10] S. Goldin-Meadow, "The role of gesture in communication and thinking" in Trends in Cognitive Sciences 3(11), pp. 419-429, 1999.

[11] J. Cassell, D. McNeill, K. McCullough, "Speech-gesture mismatches: Evidence for one underlying representation of linguistic and nonlinguistic information" in Pragmatics & cognition 7(1), pp. 1-34, 1999.

[12] N. Chafai, C. Pelachaud, D. Pele, "Towards the Specification of an ECA with Variants of Gestures" in Intelligent Virtual Agents, pp. 366-367, IVA 2007.

[13] S. Prillwitz, "HamNoSys Version 2.0: Hamburg notation system for sign languages: An introuctory guide. Signum", 1989.

[14] L. Rai, B.H. Lee, J. Hong, H. Hahn, "Intelligent gesture generation in humanoid robot using multi-component synchronization and coordination" in the ICROS-SICE International Joint Conference, pp. 1388-1392, 2009.

[15] H.H. Kim, H.E. Lee, Y. Kim, K.H. Park, Z.Z. Bien, "Automatic Generation of Conversational Robot Gestures for Human-friendly Steward Robot" in Robot and Human Interactive Communication, pp. 1155-1160, ROMAN 2007.

[16] A. Hiraiwa, K. Hayashi, H. Manabe, T. Sugimura, "Life size humanoid robot that reproduces gestures as a communication terminal: appearance considerations" in Computational Intelligence in Robotics and Automation, pp. 207-210, 2003.

[17] R. Gelin, C. d'Alessandro, Q.A. Le, O. Deroo, D. Doukhan, J.C. Martin, C. Pelachaud, A. Rilliard, S. Rosset, "Towards a Storytelling Humanoid Robot" in AAAI Fall Symposium Series, 2010.

[18] C. Pelachaud, R. Gelin, J.C. Martin, Q.A. Le, "Expressive Gestures Displayed by a Humanoid Robot during a Storytelling Application" in Second International Symposium on New Frontiers in Human-Robot Interaction, United Kingdom, AISB 2010.

[19] M. Kipp, "Anvil - a generic annotation tool for multimodal dialogue" in 7th Conference on Speech Communication and Technology, 2001.

[20] Q.A. Le, C. Pelachaud, "Generating co-speech gestures for the humanoid robot NAO through BML" in International Gesture Workshop, Athens, pp. 60-63, GW 2011.

[21] J. Monceaux, J. Becker, C. Boudier, A. Mazel, "Demonstration: first steps in emotional expression of the humanoid robot Nao" in the conference on Multimodal Interfaces, ICMI-MLMI 2009.

[22] A. Beck, A. Hiolle, A. Mazel, L. Canamero, "Interpretation of Emotional Body Language Displayed by Robots" in the 3rd international workshop on Affective interaction in natural environments, AFFINE 2010.

[23] B. Mutlu, J. Forlizzi, J. Hodgins, "A Storytelling Robot: Modeling and Evaluation of Human-like Gaze Behavior" in the 6th IEEE-RAS International Conference on Humanoid Robots, Genova 2006.

[24] J. Li, M. Chignell, "Communication of Emotion in Social Robots through Simple Head and Arm Movements" in International Journal of Social Robotics, Springer, 2009.

[25] M. Bennewitz, F. Faber, D. Joho, S. Behnke, "Fritz - A Humanoid Communication Robot" in the IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2007.