DR 5.2: Robot and avatar behavior implementation

Bert Bierman, Mario Fumagalli, Diego Fumagalli

Produxi. Mixel

‹bert@produxi.nl, m.fumagalli@mixelweb.it, d.fumagalli@mixelweb.it›

Project, project Id: EU H2020 PAL / PHC-643783

Project start date: March 2015 (48 months)

Due date of deliverable: M24

Actual submission date: M24

Lead partner: Produxi

Revision: draft

Dissemination level: PU

In this document we report on the work done in WP5 as part of deliverable D5.2, in particular task 5.2. Task 5.2 focuses on actual implementation of behaviors (e.g., the actual specification of the gestures) on the physical and virtual NAO.

Executive Summary

In this document we report on the work done in WP5, specifically task 5.2. It consists of creating the module to generate the actual (virtual) robot (NAO) behaviors. An important objective is to develop a framework that uses a common (1) knowledge-base, (2) gesture & posture creation method and (3) gesture & posture expression space for both the robot and its avatar. The module converts gesture specifications ("commands") into values about joints and times to be send to the robot or its avatar (i.e., it can communicate gesture & posture commands to both the robot and its avatar). The current framework meets two important challenges for cross-platform social human-agent interactions. First, the behaviors of the virtual and physical NAO have completely the same foundation and expression mechanisms, so that can be perceived as really similar. Second, emotional (mood) modulation of these gestures are possible. Mood modulation of gestures was based on previous work and extended to also work with the virtual NAO.

1 Tasks, objectives, results

The overall aim of the project is to develop a Personal Assistant for healthy Lifestyle (PAL), a system that will assist the child, health professional and parent to advance the self-management of children with type 1 diabetes aged 7 - 14, so that an adequate shared patient-caregiver responsibility for child's diabetes regimen is established before adolescence. PAL will be composed of a social robot, its (mobile) avatar, and an extendable set of (mobile) health applications, which all connect to a common knowledge-base and reasoning mechanism.

This deliverable reports the work on the generation of behaviors of the social robot and its (mobile) avatar. These activities are embedded inside the PAL system as shown in figure 1.

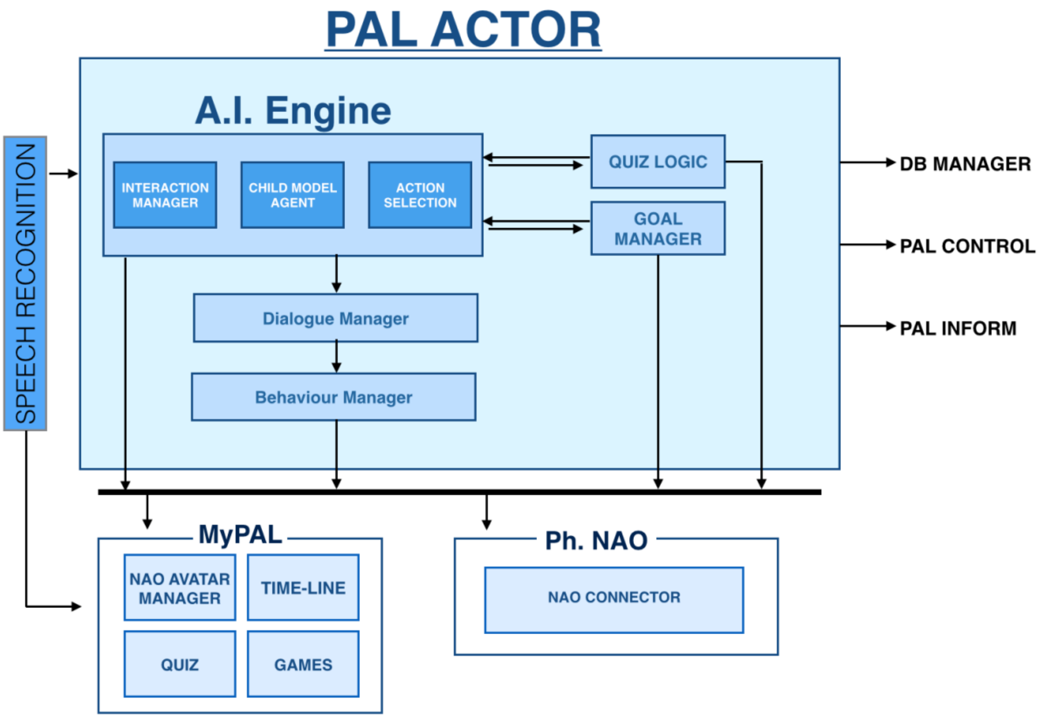

Figure 1: The relations between de various components of the PAL system

The *Behavior Manager* listens to messages of the Nexus containing specifications of multimodal (utterances). These utterances are converted into gestures which are actual executed on the NAO (both virtual as physical) by the NAOConnector. This report describes the development of a framework for this gesture generation, which uses a common (1) knowledge-base & message board, (2) gesture & posture creation method and (3) gesture & posture expression space for both the robot and its avatar. For this, two important challenges for cross-platform social human-agent interactions have to be met. First, the behaviors of the virtual and physical NAO should have completely the same foundation and expression mechanisms, so that can be perceived as really similar. Second, emotional (mood) modulation of these gestures should be possible.

1.1 Planned work

The physical personal Embodied Conversational Agent will be a humanoid NAO robot from Aldebaran (http://www.aldebaran.com/) in physical form and in virtual form from Mixel. This task includes implementation of the behaviors needed for the core PAL agent (and potentially activities) on the NAO platform and its virtual avatar. This task is strongly related to Task 4.2, in which multimodal expression generation is researched including the use of affectively colored gestures, based on previous work [3]. While in task 4.2 the focus (with respect to the NAO) is on how to generate gestures that can be affectively modulated and can be of support to the dialog. Here the focus is on actual implementation of the behaviors (e.g., the actual specification of the gestures). We also build upon the DUT RoboTutor work, enabling scripted robot-to-audience behaviors, but will need to extend this. In particular, to be able to use a NAO, a Behavior Generation Framework needs to be developed which facilitates interaction between the NAO platform and the core PAL agent. Further, we envision two types of NAO behaviors: a) functional behaviors that are specific for the core mHealth applications in which the NAO is going to participate, like a quiz or a sorting game and b) nonfunctional behaviors and behavior modulations that make the NAO to appear “alive” (e.g., move the arms, legs and head in small (random) steps and blink with its eyes, Perlin noise), and to express emotions and moods.

1.2 Actual work performed

We report here on the three modules of the work in WP5: the development of the MyPal-app, the BehaviorManager and the NAOConnector.

Figure 2 shows how the different parts are connected.

The physical agent is Aldebaran’s NAO. For the virtual agent a model of that NAO has been created in Unity and software functionality is added to enable moving the model by commands from the system. The virtual NAO and the physical NAO receive exactly the same information from the Behavior-Manager to enable identical behavior.
Both are using the same text-to-speech engine to assure the voices sounds identical. For the Italian language Mary TTS is used, but because there is no Mary TTS support for Dutch, the cloud services of ReadSpeaker (www.readspeaker.com) are applied in the Netherlands. Components are created in both MyPal-app and NAO to enable to handle the text-to-speech conversion.

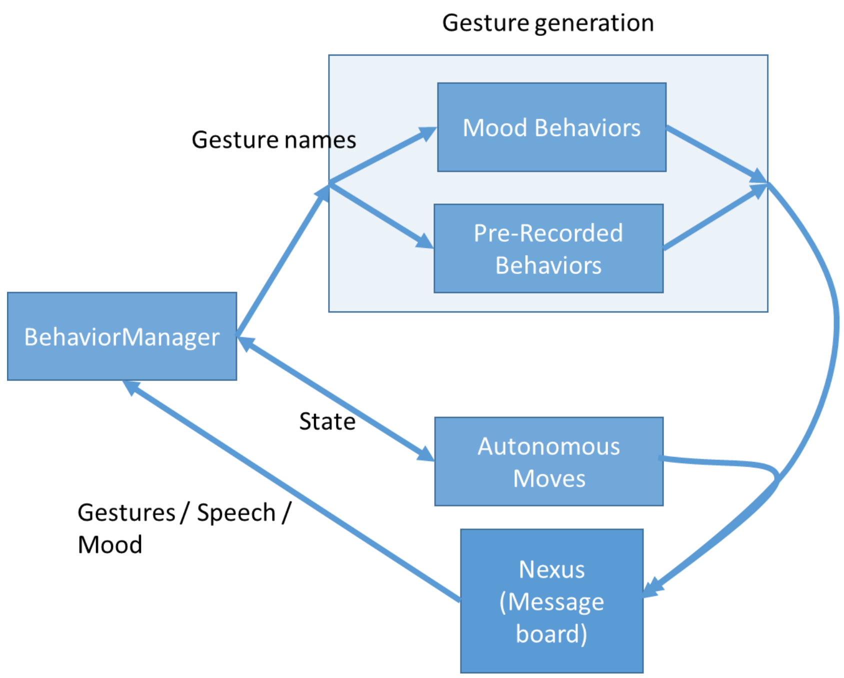

Figure 2: Graphical representation of relations between the PAL messaging system, the BehaviorManager

The BehaviorManager

The BehaviorManager module is responsible for the generation of the movement of the (virtual) NAO. The messaging system (Nexus) is a publish/subscribe mechanism. The BehaviorMessage subscribes to the messages which contain the multimodal utterance. The utterance consists of the name of the gestures to execute and the text to be spoken as well as the emotional state of the child. From this content it constructs the necessary values to move the joints of the (virtual) NAO which are then published to the Nexus.

The task of the BehaviorManager can be divided into two parts i) make the (virtual) NAO look lively and ii) make the make the (virtual) NAO execute the multimodal utterance.

Make the (virtual) NAO look lively

The activity of this part is to realize a (virtual) NAO which is looking lively (do autonomous moves). When the (virtual) NAO is looking alive it will be more engaging for the child. These autonomous moves need to be carefully combined with the deliberate movements resulting from Behavior messages from the PAL

system.
Currently the autonomous movements are sending randomly modulated signals to the head, the arms and the hips/legs, as well as blinking with the eyes.

This autonomous moving is applied when the (virtual) NAO is doing nothing. When the NAO is doing something, but not indicated with gestures (so like only talking) several other procedures are applied. There are so called conversational fillers, which implement gestures to visualize the NAO is “thinking” before giving an answer [6].
The last set of gestures is applied when the (virtual) NAO is talking. During this talking the arms are moved in a random way to resemble the arm movement people use when talking.

The gesture generation

The BehaviorManager receives a multimodal utterance which contains a name of a gesture, the text to be spoken and the emotional state of the child (derived by the ChildModel) The value is used to modulate the actual gesture the actor displays. The actor in this ways mirrors the emotional state of the child. From the gesture name the BehaviorManager needs to calculate the position of the joints of the (virtual) NAO and the time available for the joints to reach that position. These conversions are defined in different ways: i) mood behaviors and ii) pre-recorded behaviors.

i) Mood behaviors.
Mood-behaviors are defined using both joint over time values and a procedure to modulate the values by an emotion parameter in terms of valence and arousal [1, 2, 3].

In the first procedure for each gesture a sequence of postures is defined. Based on the emotional parameter the postures are adapted with regards to the amplitude of the posture and the speed with which the sequence is executed. The resulting joint and time values are published and received by the NAOConnector.
These behaviors originate from the DUT RoboTutor. This mechanism has been tested for correct recognition of affect in the gestures as well as effects on observers in several usage scenarios [4, 5]. Since the RoboTutor is created using C#, a programming language no longer supported by the NAO, the source code has been ported to Java. The rough conversion is done with the CS2J tool from Twiglet Software after which the remaining changes where done by hand. After that, the resulting code is integrated into the workflow of the already available BehaviorManager.

In the second procedure also for each gesture a sequence of postures is defined with average amplitudes. These postures can be combined which results in a more comprehensive gesture. The given amplitude of the joints in the gesture is modulated by the mood value (a value between -1 and 1) from the child model. The modulation at each time point is calculated with:

$$value_{(t=t)} = value_{(t=0)} + \text{mood value} * (value_{(t=t)} - value_{(t=0)})$$ [4].

The result is that with higher mood value the amplitudes will be larger and with lower mood values the amplitude will be lower.

iii) Pre-Recorded behaviors.

These behaviors are situated on the NAO and mostly contain complex gestures like stand-up and sit-down. The BehaviorManager publishes the name of the gesture to be executed and the NAOConnector processes that message.

The NAOConnector

This module is responsible to have the (virtual) NAO move and/or speak. It subscribes to messages containing the actual values of the joint positions and the time to reach the position, and the text to be spoken (published by the BehaviorManager). The information received is converted into actual commands of the NAO's internal system. The components involved and their relation are shown in figure 3.

This module has two different implementations:

- An implementation which is running on the tablet to communicate with the virtual NAO
- An implementation which communicates with the real NAO

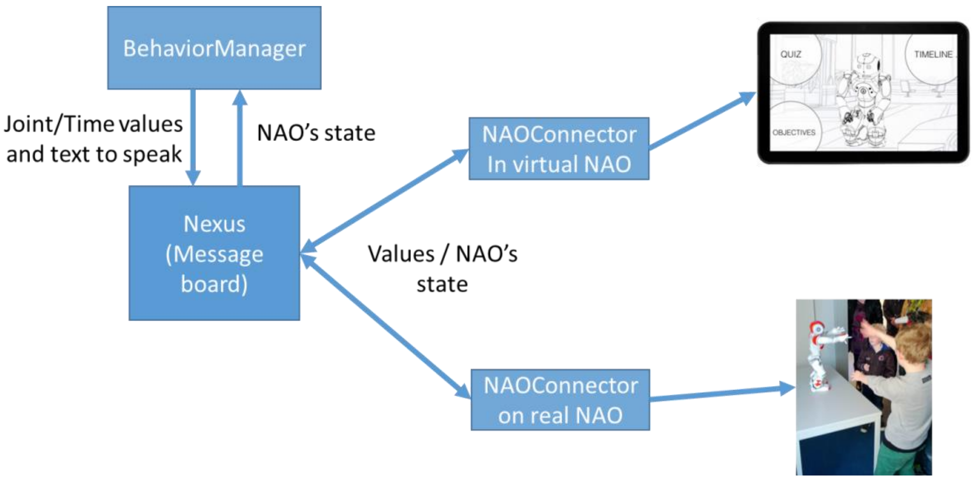

Figure 3: Graphical representation of relations between the BehaviorManager and the embodiments.

The virtual NAO

The implementation for the virtual NAO is embedded in the Unity environment in which the MyPAL application is developed and the task is to execute the moving of the model according to the values received.

Mixel is currently working on the second version of the NAO 3D Model, to solve the issues that came to light after the release of the first version.

NAO Avatar V. 1

Features

- The first version of the 3D Model is composed of n virtual objects connected through a series of junctions, which are meant to take the place of Real NAO's gears, as displayed in Figure 4 below.
- Mixel developed an algorithm to translate the rotations of the Real NAO's gears in rotations of the NAO Avatar's junctions around their three axis's (x, y, z), so that we can obtain the same movements using the same commands via Behavior Manager or via NAO Connector.
- The user can choose amongst different colors for the NAO Avatar: blue, red, orange, green, pink and white. See figure 5.

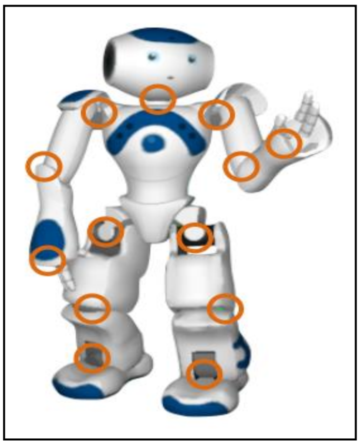
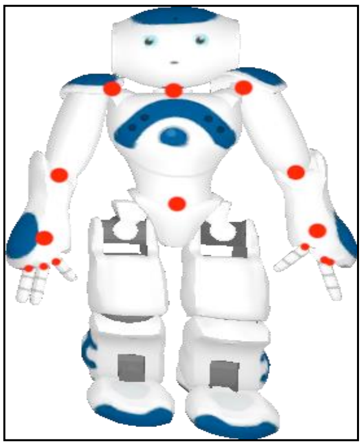

Figure 4: The real NAO on the left with the joint indicated and the avatar on the right with the junctions indicated which were implemented.

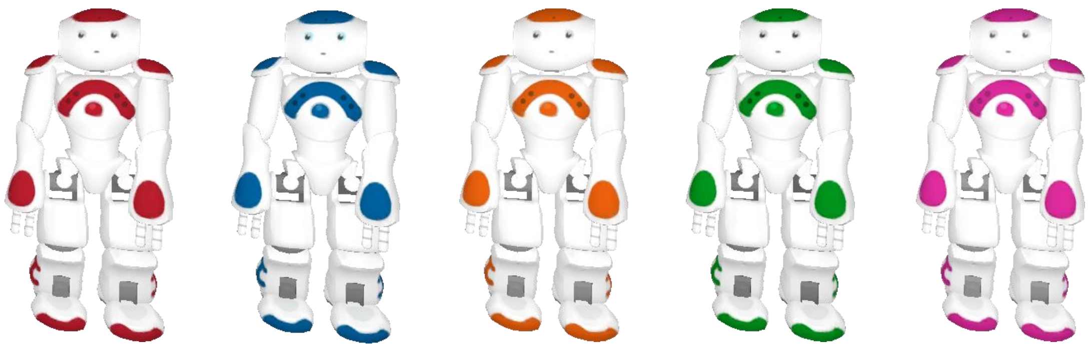

Figure 5: Colors of the NAO avatar which could be used in the MyPal app.

The main problems spotted with the first release of the Avatar were the following:

- The junctions were not all placed correctly, with consequent issues in the movements output;
- The techniques we used to design and develop the Avatar force its physical main anchor (its pivot) to be static. This design implies that when the Avatar goes for instance into crouch position his head remains on the same height but the knees and feet are raised. So the Avatar is no longer having its feet on the floor, but seems to float in the air.

NAO Avatar V. 2

To solve the problems experienced with the Avatar's movements, Mixel started working on a new release.

The new version's features will be the following:

- A higher number of polygons and vertices, to make the Avatar more similar to the Real NAO;
- Glowing eyes to obtain a slightly more accurate "facial" expression;
- Giving NAO Avatar a correct and realistic legs movement still represent the most difficult work section. There are two ways to work around this issue:
    - Gravity simulation: simulating a working "gravity" that forces the Avatar's body and drags it down when flexing its knees was the first attempted solution, and it gave some good results. The issue we got with this solution is the subsequent need to insert and apply a new set of parameters for each and every component.
    - Inverse kinematics: using inverse kinematics process to achieve the right junctions' angulation is a known solution in this field, but it is impossible to use in this case since there is no system to calculate, correctly manage and realistically simulate the position of the center of gravity and its path through the Avatar.

  Mixel has been working on a hybrid solution, one that allows us to calculate only a portion of the parameters required to apply a simulated gravity to some of the modules. This solution should be completed by January 2017.

The real NAO

The implementation for the real NAO is a wrapper which converts the data received (from the BehaviorManager) into the commands which can be send to the NAO's own execution system. The majority of the messages originate from the autonomous move system. The main issue to solve is handling all messages in parallel without blocking the processes.
The module also monitors if an execution and/or speech is finished, if its

sensors are touched and processes the sound localization messages. This information is published to the PAL system.

The NAOConnector can run on the server or on the NAO itself. The advantage of use it on the NAO is a reduction of the amount of data to be send over the network and it simplifies the registration process during the experiment. Disadvantage is the NAO needs to be equipped with the Java run time environment, which is not installed by default.

2 Conclusions

A framework has been developed for the generation of behaviors for the PAL robot and its avatar, which uses a common (1) knowledge-base & message board, (2) gesture & posture creation method and (3) gesture & posture expression space. The framework converts gesture specifications ("commands") into values about joints and times to be send to the robot or its avatar (i.e., it can communicate gesture & posture commands to both the robot and its avatar). Two important challenges are met for cross-platform social human-agent interactions. First, the behaviors of the virtual and physical NAO have completely the same foundation and expression mechanisms, so that can be perceived as really similar. Second, emotional (mood) modulation of these gestures is possible.

Major Outcomes

- A framework has been developed which contains:
    - a module that applies the same knowledge-base and controls the behaviors of the NAO-robot and its avatar with the same methods, so that the robot and avatar show completely similar behaviors in the concerning conditions,
    - an implemented method for the creation of a lively (virtual) robot that can express moods
    - an implemented mechanism to adapt the PAL actor mood to the mood of the child
- The behavior of the avatar is greatly improved with respect to the previous one. Specifically the number of joints implemented is increased and the already present joints are better positioned. With these changes the ability to use hands and legs is added, and the resemblance of the arm movements by avatar or NAO is enhanced.

3 References

1. Tielman, M. Expressive behaviour in robot-child interaction, Master Thesis, 2013.
2. Tielman, M., Neerincx, M.A., Meyer, J.-J. Ch. And Looije, R. (2014). Adaptive Emotional Expression in Robot-Child Interaction. In Proceedings of the 2014 ACM/IEEE international conference on Human-Robot Interaction. ACM, New York, NY, USA, 407-414.
3. Xu, J., et al., *Mood Expression through Parameterized Functional Behavior of Robots*, in *RO-MAN 2013*. 2013, IEEE. p. 533-540.
4. Xu, J., et al., *Bodily Mood Expression: Recognize Moods from Functional Behaviors of Humanoid Robots*, in *ICSR 2013*. 2013.
5. Xu, J., et al., *Robot mood is contagious: effects of robot body language in the imitation game*, in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 2014, International Foundation for Autonomous Agents and Multiagent Systems. p. 973-980.
6. Wigdor, N., de Greeff, J., Looije, R., & Neerincx, M. A. (2016). How to improve human-robot interaction with Conversational Fillers. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on* (pp. 219-224). IEEE