Seventh Framework Programme
Theme ICT-2009.2.1
Cognitive Systems and Robotics

HUMAVIPS
Humanoids with Auditory and Visual Abilities in Populated Spaces
Grant agreement number 247525

D1.3: Scenario and Detailed Specification of M24 Demonstrator

Date of preparation: 29 July 2011

Contributors:
Pierre-Emmanuel Viel, ALD
Jan Cech, INRIA
Xavier Alameda-Pineda, INRIA
Daniel Prusa, CTU
Vassil Khalidov, IDIAP
Jean-Marc Odobez, IDIAP
David Klotz, BIU
Johannes Wienke, BIU
Britta Wrede, BIU
Sebastian Wrede, BIU

Contents

1 Scenario Introduction

To demonstrate our year 2 and year 3 results, we have decided to choose a "Vernissage" scenario where Nao acts as a guide robot in a small art gallery taking care of a corner of the room where several paintings are exhibited. The robot is waiting in front of the paintings, being aware of the visitors in the room. When a visitor or a group of visitors gets within the interaction range of Nao, a contact is established and Nao initiates the interaction by offering to provide information about the paintings. One important issue in our scenario is the challenge to address multiple listeners as well as to check their attention during the explanation.

Although this scenario appears very similar to already existing museum guide robots, it pushes the envelope of current state-of-the-art perception and action capabilities of social robots by applying research results from HUMAVIPS challenges. Thus, the HUMAVIPS guide robot will be able to monitor its environment especially with respect to humans that may want to or are interacting with it and react in a much more contingent way than existing museum robots.

Our goal is to enable coherent human-robot interaction in a highly complex situation with several visitors appearing in the scene, thus offering the opportunity to tackle the "cocktail-party effect", a problem for which we try to build algorithms that mimic the human ability to focus attention on just one person in the midst of other people, voices and background noise. Figure 1 shows the robot in another art exhibition scenario that exemplifies many of the challenges we also want to address in the scenario specified in this document.

Nao's perception of the world relies on vision and auditory algorithms. Information required to monitor and correctly interpret state and events in the vernissage room relates mostly to humans. Many attributes are needed to be estimated or tracked for each person (identity, 3D location, age, gender, head pose, gesture, talk, etc.). Standalone algorithms processing visual or audio data can be proposed to fulfill the inducted particular tasks. However, to achieve a really robust perception, it is necessary to make decisions on the basis of multi-modal (visual and auditive) information, hence, taking the advantages of data redundancy and complementarity.

For example, the auditory channel is relatively independent of direction and provides thus a good means to shift the robot's attention when needed. In our scenario this means that the robot will be able to search for interaction partners not only visually but also auditorily by e.g. searching for foot-steps or other characteristic human noises, which will trigger it to turn around for visual search. This is an example of complementarity, a feature we want to explore in more depth in the envisioned scenario. Redundancy is necessary in the current scenario to tackle the problem of robust person tracking. Similarly timeliness and costs of information will be taken into account in the fusion process of multi-modal information as proposed by [15].

The development of the scenario for year 2 will be based on our achievements in year 1 where we already provided first bilateral integrations (e.g. Visual focus of attention and engagement tracking and dialog management in [12]). This document describes at different levels of details the further integration of developed algorithmic solutions in the overall robot system with a focus on results to be achieved in year 2. Our restrictions for year 2 concern the number of persons in the room which we will initially restrict to a small group of at most five persons. Other criteria are the way how a group of people can interact with the robot (e.g. they do not speak simultaneously) or a knowledge of the exhibition environment setup (e.g. positions of the paintings). Movement of the robot will in year 2 be restricted to phases where no or only limited perception has to be performed. We aim to relax certain restrictions such as number of persons or background noises for year 3 which will be specified in the next year's deliverable.

The content of this deliverable is organized as follows. A detailed scenario environment specification is given in Section 2. Section 3 analyzes different situations that the robot can encounter and outlines how it should behave. A decomposition of the scenario into logical units is listed there. Functional modules needed to implement the whole scenario are explained in Section 4. Software architecture and integration specific details are discussed in Section 5. Evaluation strategies that will be employed on different levels are specified in Section 6. Finally, a schedule for an incremental development strategy, including milestones, is set out in Section 7.

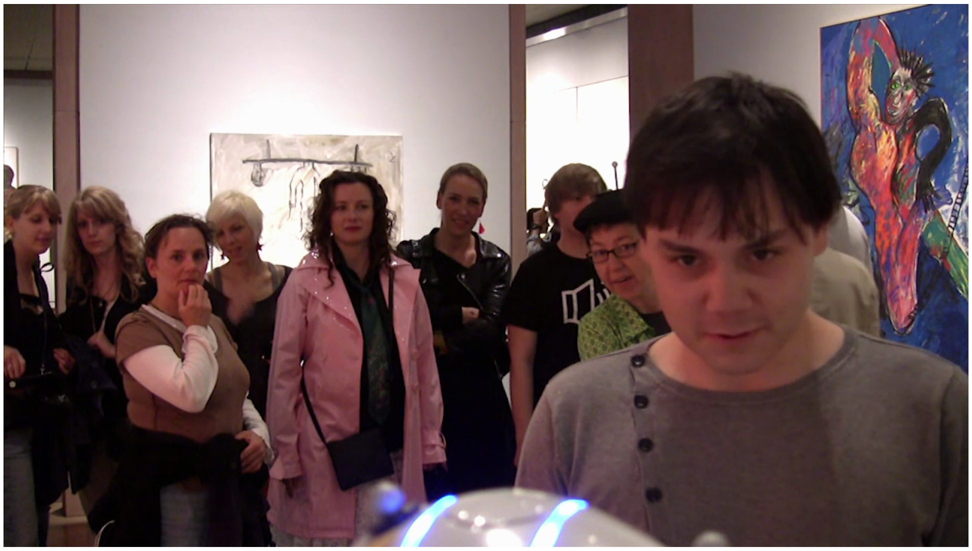


Figure 1: Nao interacting with a group of museum visitors (from data collected for [21]).

2 Environment Specification

This section introduces the environment in which the scenario will take place and describes restrictions which will be made in order to create a realizable situation. The overall setting is an exposition room in a small art gallery or a museum, specified in a way that it is possible for every partner to set up a version of the environment at the respective labs. In the room several paintings (or similar exhibits like scientific posters in the lab version) are exhibited on two walls. Nao walks on a table in front of the exhibits to compensate its low height, but if feasible, the ambition for year 3 is to remove the table.

The remainder of this section starts with a more detailed specification of the room layout in Subsection 2.1. The lighting and acoustic conditions of the room used for the scenario are described in Subsections 2.2 and 2.3 respectively. Finally, some constraints on the interaction of humans with the robot are detailed in Subsection 2.4.

The partners will follow all given specifications as close as possible while setting up their version of the environment. However, small variations will be possible, since implemented algorithms have to be robust enough to compensate for these variations.

2.1 Room Layout

The room for the setup will be in a normal size e.g. of an office for one or two persons. A detailed specification of its layout is shown in Figure 2. The table on which Nao walks is shown in gray and the paintings are shown with red lines, dimensions are displayed. The heights of the table and of the paintings are 0.75 m and 1 m respectively and the height of the gap between the table surface and the lower edge of the paintings is 0.3 m. We keep the possibility to add artificial markers to the scene (around the table) for a baseline navigation approach.

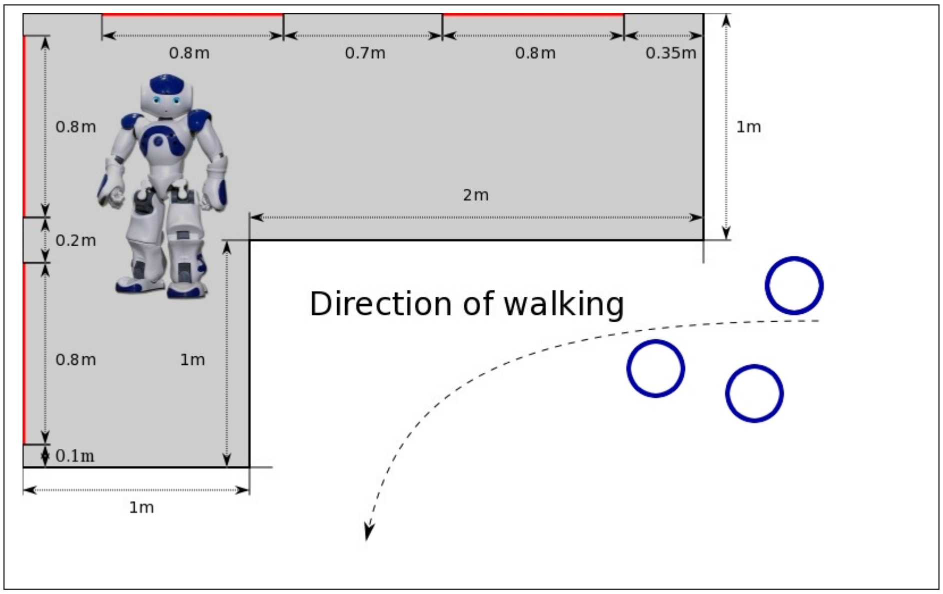


Figure 2: Physical specifications of the scenario. The table on which Nao walks is shown in gray, the paintings are depicted in red color. Approaching people are indicated as blue circles.

2.2 Room Lighting

No special modifications will be done to the room. The only acceptable modification is adding diffuse light to account for good lighting conditions. Optionally we will install spot lights aiming at the paintings. This modifications may be necessary to perceive good visual features on the paintings and also fits into the overall scenario.

2.3 Room Acoustics

The developed algorithms aim at settings in small rooms without harsh acoustic conditions like in great halls. Therefore a carpeted office without strong reverberations is an appropriate setting that matches the small vernissage topic. There will be no special modification of the room acoustics, as e.g. attenuation reverberation, isolating materials in walls and doors, etc. Nevertheless, we will exclude outdoor and background noise like streets, music, radio, etc.

2.4 Restrictions on Human-Robot Interaction

Robust human-robot interaction (HRI) is a principle aim in HUMAVIPS. Nevertheless, a completely unstructured setting in the given scenario is not realistic, especially in year 2. In the following, the restrictions on how humans interact with the robot are thus made explicit.

The first important decision is that the group of people interacting with the robot is limited to a size of at most 5 persons. This reflects the size of the room, limits the algorithmic effort to a reasonable amount, and provides a setting where recording for the year 2 dataset is still possible. We exclude children for being able to assume an approximate height of all humans and the speech recognizer (which is not part of the active development in HUMAVIPS) is not trained for children. People interacting with Nao will be in a suitable distance for algorithmic processing that models the human habits als close as possible, e.g. about 1 to 2 meters. We will evaluate an appropriate range with the availability of the new head which contains new cameras. The range will reflect a tradeoff between the visible area depending on the field of view and the required resolution for image processing. The chosen setting with a table will provide better views on the people and the overall scene for perceptual processing. Nevertheless we will evaluate if and how much the camera system of the new head for Nao simplifies the perceptual tasks in a setting without the table.

To reduce the complexity of speech recognition, each human pariticipant of the scenario will use a headset. The headsets will only be used for this purpose and will perspectively be removed as soon as other supporting modules like the sound source separation are available.

People in the scene will be instructed not to speak simultaneously or while the robot is talking. Moreover, occasional eye contact with the robot is required and special instructions will be given if really necessary.

3 Scenario Decomposition

The achievement of the proposed scenario is a complex problem that relies on several interleaved modules fulfilling different functionalities, from sensing the world in order to extract the relevant information to behavioral control of the robot. These modules are used and should be triggered whenever necessary, according to the context.

From the perspective of modeling the robot's behavior, the scenario continuously unfolds in different stages that involve changes in both its internal world representation and its behavior. These stages are the result of its understanding and recognition of different typical situations that happen in sequence or parallel and that move the robot state forward. The goal of this Section is thus to perform a finer analysis and decomposition of the scenario, and, before dwelling into the functional details and specification of the robot's software components, to identify the different situations that the robot will or might encounter within the scenario, and what we envision how it should behave in such cases.

In the next subsection, we start by introducing general principles related to the robot's perception and its architectural organization. Next, we present a more detailed analysis of the robot's dynamic behavior in typical situations in the given scenario. Please note that a more detailed break-down of these general principles and the outlined activities into separate functions and eventually software components is presented in Sections 4 and 5.

3.1 Overarching Principles

The stance we take in HUMAVIPS on Human-Robot Interaction is grounded on the fact that multi-modal (i.e. visual and auditive) information is a necessary prerequisite for robust perception by taking advantage of redundancy and complementarity. In addition to this, the robot behavior needs to reflect the fact that the environment is changing – and so is its perception and its reliability – and must thus allow for mixed-initiative, that is allow the user to initiate an activity but also to initiate a behavior based on internal processing results (e.g. when the user can not be perceived any more).

To facilitate smooth integration of both perception and production, we need an architectural organization that allows on the one hand for light-effort software integration but that also takes care of unifying different representations originating from different modules (e.g. ego- vs allocentric data), tackles asynchrony and supports principled system evaluation. We also try to motivate some general technical principles realized in the demonstrator at this point in the document since they have a strong influence on the overtly visible behavior of the robot.

Perceptual modules. A great part of the software modules relates to the perception of humans in the world. In particular, since in HUMAVIPS the emphasis is on interacting with people, the majority of the perception modules have the goal of maintaining the robot's representation of the world on this aspect.

This encompasses many things. At a first level, detecting how many people are there, and for each of them (according to their interest for the robot, cf. dialog and engagement modeling) extracting his/her physical and visible attributes: where is the person, who is the person (what is his face or voice, how is he dressed), what is he doing (where is he looking at, speaking status, head gesture). These attributes convey both, complementary as well as redundant information and will be combined accordingly into consistent person hypotheses that are maintained over time. In general, most of the modules involved in this task run in the background to continuously update a person's representation maintained by a dedicated mechanism responsible for *Person Tracking.* At a second level, this involves the understanding of these attributes in the form of more general concepts like the intention a person or group may have to interact with the robot or the level of interest shown towards the explanations given by the robot. These mechanisms are summarized in the following sections under the name *Engagement Tracking.*

An additional overarching design decision is the choice of the reference space in which the information of robot localization modules is stored. In HUMAVIPS, we adopt an ego-centric approach in which all variables refer to a robot coordinate system as it eases the development of robot behaviors for social human-robot interaction which often requires only distance and azimuth information of persons or objects such as paintings in the scene.

Note however the following. A certain amount of location information is given to (or learned by) the robot in a room-referential coordinate system (positions of doors, of paintings, visual landmarks) forming a map of its environment. The robot will perform localization tasks based on this map and will maintain the coherency of its ego-centric representation according to it.

Architectural Organization

From an architectural perspective the system will be built with the principles of component-based composition and event-driven architectures. We adopt the definition of [25] that states:

> A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.

An appropriate middleware manages the event-based communication between components with support for efficient transmission either over the network or in-process, to provide a sufficient performance for the perceptual algorithms in the project. Further features that support the development cycle of a system like this are integrated as well.

The system components operate on different levels of abstraction. Starting with modules that interface the robot's sensing and actuation capabilities for the architecture and continuing with perceptual processing for visual and auditory cues. At the highest level modules for behavior and dialog management will control the envisioned scenario. Defined sets of exchanged events between components will be made persistent for a short-term period of time by special memory components in the software architecture. On the one hand, this will support synchronization between asynchronously operating components, and on the other hand will provide advanced subscription and query support for extracting temporally defined events, e.g. to detect conditions on the trajectories of persons in the scene accounting for their temporal dynamics. The system will be deployed to a set of external computer which will support the limited computing power of Nao and most scientific components will operate on these external computers. The envisioned architectural strategies are described in more detail in Section 5.

3.2 Modeling the Robot's Activities

The remainder of this Section will analyze and decompose the proposed scenario into smaller sub-scenarios describing the necessary robot behaviors on a more detailed but still abstract level. These sub-scenarios are represented as structured activities which the robot system has to carry out in interaction with one or more humans. As such this section contributes as part of the requirement analysis for the overall system by specifying the visible result of year 2.

The following sub-scenarios are referenced throughout the whole deliverable, e.g., to describe which functional components jointly realize a complex functionality or to contextualize requirements on the resulting software modules as well as on the environment. In order to model the logical activities within the different functional subsystems and their interactions, we will employ activity diagrams as a visual notation, defined in the UML2 (Universal Modeling Language 2 [19]) specification. Subsequently, we will explain all identified sub-scenarios in prose and discuss several relevant sub-scenarios with the help of activity diagrams. This is done in order to exemplify how we want to model and further develop the complete scenario at this level of abstraction during the course of year 2. Please note that these diagrams display only the minimal subset of the logical system activity required for the implementation of the corresponding sub-scenario to keep them as focused as possible.

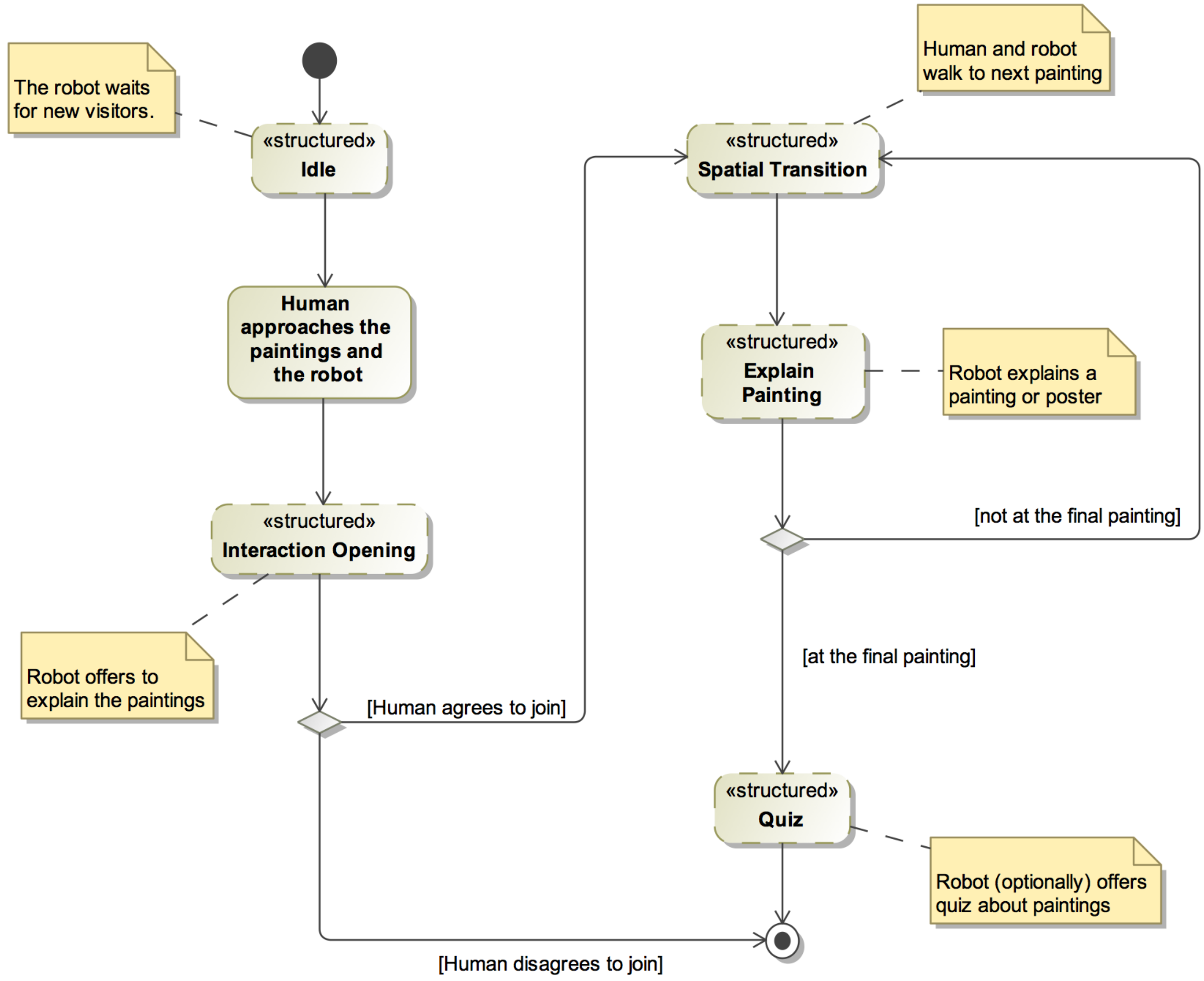

The robot waits for new visitors.
«structured» Idle
Human approaches the paintings and the robot
«structured» Interaction Opening
Robot offers to explain the paintings
[Human agrees to join]
[Human disagrees to join]
Human and robot walk to next painting
«structured» Spatial Transition
«structured» Explain Painting
Robot explains a painting or poster
[not at the final painting]
[at the final painting]
«structured» Quiz
Robot (optionally) offers quiz about paintings

Figure 3: Activity overview. Structured activities are detailed in the following sections.

Analyzing the complete scenario as described in the beginning, we decomposed the overall scenario into sub-scenarios which will be explained in a sequence according to an ideal walk-through through the "Vernissage". Naturally, deviations of this strict sequence are bound to occur, e.g., according to perceptual uncertainties which we try to cope with either directly in the different algorithmic functions (to be explained in Section 4) or by explicitly modeling these deviations already at this level of abstraction in the flow of activities. Basically, the goal of the behavior of the robot to be achieved in year 2 is to complete a whole walk-through of a person through the four paintings and to offer a quiz (e.g. about the paintings) at the end of the tour. This provides sufficient complexity to approach the scientific challenges motivated in Section 1, e.g. the continuous tracking of the visitors' engagement levels. Potential add-ons to this basic functionality are given throughout the descriptions in terms of extensions and at the end of this section. More possible extensions will arise during the realization of the described behaviors. Figure 3 shows an overview of the general activity flow in this scenario.

3.2.1 Idle

This activity is a description of the initial idle state the robot will be in at the start of the scenario. In this state, the robot optionally moves to (or starts at) a designated "home" position (e.g. in front of the first of the painting) and waits for new interaction partners.

When there are new visitors within a pre-specified range of the robot, the robot will try to open an interaction with them, cf. the respective activity in Section 3.2.2. In later iterations of the scenario this could also include actively searching for new interactions partners, e.g. following the behavior described in Section 3.2.6.

3.2.2 Interaction Opening

This activity describes how the robot and different parts of the larger system controlling it will behave when trying to open an interaction with newly arrived visitors in the exposition room. Figure 4 describes the general activity flow in this situation, which is explained in the following paragraphs.

Note that different parts of the robot's perception and the functionalities that build on them (like the person tracking or the engagement tracking) are continuously running during the whole course of this activity and their results influence the flow of events at several points. The structured activity *Check Listener* (see Section 3.2.7) is an example of this which constantly checks if the visitors are still engaged in the interaction with the robot.

At some point, one or several of the human visitors will enter the range of the robot's sensors. The person tracking will at this point initialize a track for each of them and will continuously try to update its internal hypothesis about the visible persons. The activity *Search for Interaction Partner* (see Section 3.2.6) tries to find a suitable interaction partner for the robot and finishes when a person is found.

The *Approach Human* activity waits for such a new hypothesis and reacts to it by trying to approach the appropriate person and orient the robot towards it (this activity is explained in more detail in 3.2.8). When this succeeds and the person shows the intention to interact (as determined by the *Engagement Tracking*), the robot will offer to explain the first painting or poster. When the visitors accept this offer, the robot will try to guide them to the appropriate painting, an activity which is described in Section 3.2.3. This can be expanded to mixed-initiative interaction in later stages by also allowing the person to address the robot first.

3.2.3 Spatial Transition

Since there are several paintings or posters in the room, some process of transitioning between them while still maintaining an active interaction with the current interaction partners is needed. To keep the engagement of the visitors up, the robot will continually explain its behavior (e.g. by saying something along the lines of "Now let's take a look at X, please follow me...") while approaching the object it wants to explain next.

When the robot has moved to this new location, it has to check whether it succeeded in guiding the visitors to follow it to the next object of interest. This is done by the *Check Listener* activity described in Section 3.2.7. If the interaction partners have followed along and still show signs of engagement, the robot will start to explain the current poster or painting,

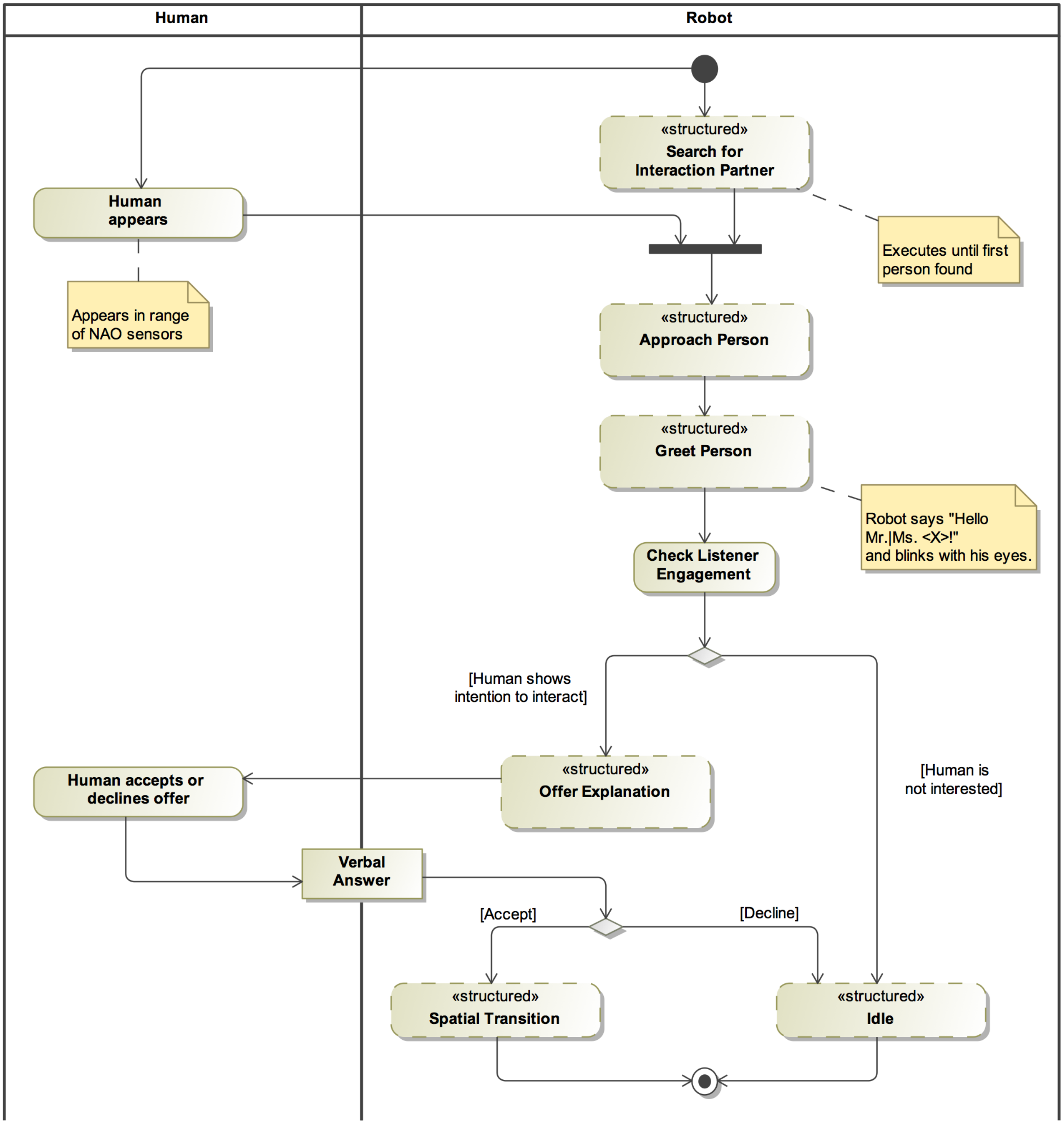

Human
Robot
«structured» Search for Interaction Partner
Executes until first person found
Human appears
Appears in range of NAO sensors
«structured» Approach Person
«structured» Greet Person
Robot says "Hello Mr.|Ms. <X>!" and blinks with his eyes.
Check Listener Engagement
[Human shows intention to interact]
[Human is not interested]
«structured» Offer Explanation
Human accepts or declines offer
Verbal Answer
[Accept]
[Decline]
«structured» Spatial Transition
«structured» Idle

Figure 4: Activity: Interaction Opening

cf. Section 3.2.4. If this is not the case, the robot has to recover the visitors by the process described in Section 3.2.6. In later stages of development, this activity could be expanded by continuously checking the visitors engagement while moving from one object to another.

3.2.4 Explain Painting

As a core activity of the "Vernissage" scenario, the robot has to explain a painting to one or more visitors. It will do this by delivering a set of predefined explanatory sentences, some of which will be accompanied by appropriate gestures (e.g. pointing out specific parts of a painting). The explaining can also include questions to the interaction partners as a means to keep them engaged, e.g. "Have you seen the detail Y here in this painting?". The answers given by the visitors to these questions can influence e.g. the level of detail of the following explanations.

At several points in this explanation (or optionally also continuously) the robot will check if the listeners are still engaged in the interaction by the process defined in the *Check Listener* activity (cf. Section 3.2.7). When the explanation for the current object of interest is finished, the robot will try to guide the visitors towards the next object, cf. Section 3.2.3. After explaining the last object, the robot could offer to hold a small quiz game (cf. Section 3.2.5) with the visitors about the content of the posters or paintings.

A possible extension to this activity is making it interruptible with gestures.

3.2.5 Quiz

This is a possible successor of the *Explain Object* activity (cf. Section 3.2.4) where the robot offers to quiz the visitors about facts concerning the paintings or posters present in the room. Initially the robot will choose one of the present interaction partners, turn its head towards that person and pose one of a set of predefined questions at him or her. Afterwards the robot will wait for an answer, optionally acting appropriately if no answer is received from the person for some time.

As soon as the person gives an answer, the robot will check if the answer was correct and give some appropriate verbal feedback (e.g. "Yes, that was correct!" or "No, the correct answer would have been…"). Following this a new round of the quiz will start with a new question and possibly a new interaction partner.

During set points in the quiz interaction, the robot will also check if the interaction partners are still present and engaged as is described in Section 3.2.7. As a possible extension, the robot could also check if utterances made by the interaction partners are adressed at itself and only accept utterances where this is the case as answers to a question.

3.2.6 Search for Interaction Partner

The behavior described in this activity is used in several other activities when the robot needs to find or relocate a person to interact with. To control the specific behavior shown in these different situations, this activity can be augmented by providing it with a description of the potential search space where the robot should look for persons and can be given a timeout, after which the robot will stop searching for new interaction partners.

At the beginning of this activity the robot will start looking around without leaving its current position. As a possible extension, the robot will start wandering to random positions in its interaction space (optionally constrained by the currently defined search space, which is used mostly when trying to relocate a specific person that was seen before). If a person is detected

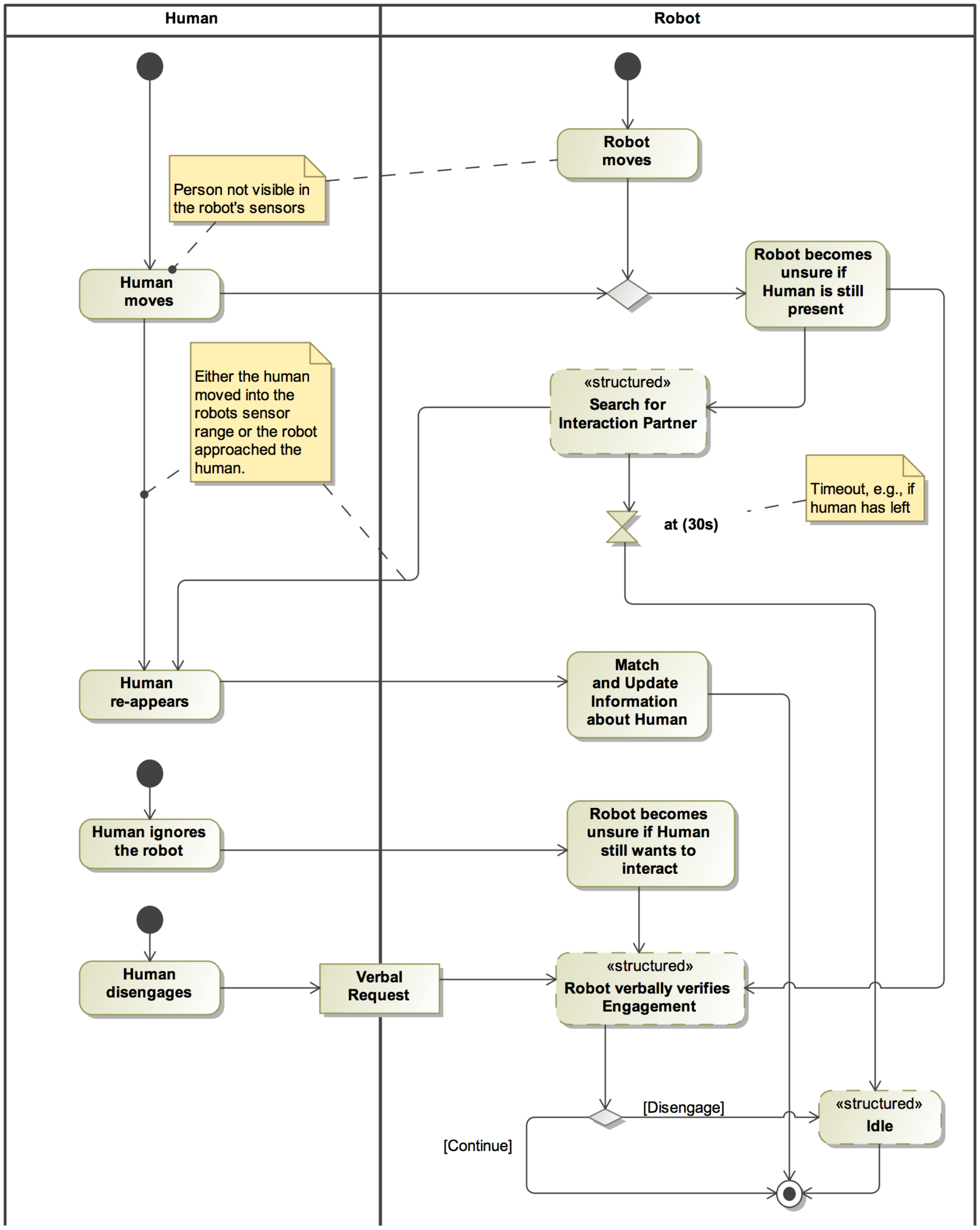

Human
Robot
Robot moves
Person not visible in the robot's sensors
Human moves
Robot becomes unsure if Human is still present
Either the human moved into the robots sensor range or the robot approached the human.
«structured» Search for Interaction Partner
Timeout, e.g., if human has left
at (30s)
Human re-appears
Match and Update Information about Human
Human ignores the robot
Robot becomes unsure if Human still wants to interact
Human disengages
Verbal Request
«structured» Robot verbally verifies Engagement
[Disengage]
[Continue]
«structured» Idle

Figure 5: Activity: Check Listener

during this search, this is used to (re-)establish a hypothesis about this person in the person tracking. As a last step in this activity the robot will approach this person (cf. Section 3.2.8) until it is in an appropriate range for interacting with it.

Another possible extension for this activity is the use of relevant information from the audio channels (additional to the visual information already used). This could be used e.g. in reacting to the detected sound of foot-steps by turning the head to the expected location where a person could be found.

3.2.7 Check Listener

This activity describes how the robot checks if visitors are still engaged in the interaction with the robot and how to react if the robot loses track of some of its interaction partners. This process is thought to be continuously running in the background as a part of several other higher-level activities (see e.g. Section 3.2.2). Figure 5 visualizes this activity.

One case where this becomes important is when a human or the robot itself moves (e.g. when moving from one painting to another one, see 3.2.3). Because the human interaction partner will probably leave the robot's field of view in both of these cases, there will be (at least for a while) no new perceptual cues that are received about this person. This in turn will lead to the respective hypothesis in the person tracking becoming ungrounded (i.e. the robot will be uncertain that the person is still there). When this happens, the activity *Search for Interaction Partner* (see Section 3.2.6 for more details) will actively try to relocate the person, starting from the last information that was known about the person's location. This is an example for the usage of top-down information influencing the lower levels of perception.

When a human re-appears (again either because the human re-entered the robot's sensor range or because the robot reoriented itself so this was the case) the person tracking should try to match this to its existing hypotheses and update the data about this person appropriately.

There are two engagement-related cases where the robot will become unsure if it should stop interacting with a person. The first case is triggered by the human ignoring the robot for a prolonged period of time (e.g. by looking in different directions and never at the robot, not responding to the robot etc.), thereby effectively causing the engagement tracking to notice that this person seems to have lost the interest in interacting with the robot. The second case is triggered by an explicit verbal request to disengage (e.g. by saying "good bye") from the human to the robot. Both cases will lead the robot to verbally verify that the human is still interested in interacting with it (this activity is described in more detail in Section 3.2.9). If this is the case, the interaction will continue normally. If not, the robot will again start behaving according to its initial *Idle* activity (see Section 3.2.1).

How to model the engagement state of a group will be subject to research during the second year. This entails how to monitor the engagment state (e.g. by level of noise or inter-group movement or by a sum of individual engagement states) and how to react to it appropriately.

3.2.8 Approach Human

In this activity the robot is approaching a potential interaction partner and trying to position itself in a good distance and orientation to the human from both a perceptual and an interactional perspective. Factors that influence this position are e.g. the position of paintings (so the robot can reference them with gestures when offering to explain them) and the ranges or resolutions of the different sensors used to build hypotheses about the interacting persons.

3.2.9 Robot Verbally Verifies Engagement

This activity is used when the robot has become unsure if the current interaction partners are still engaged and are interested in continuing the interaction. This could either be triggered by explicit disengagement actions done by the persons (e.g. saying or waving good-bye) or by a perceived lack of interest as determined by the engagement tracking (based e.g. on the fact that the persons have stopped looking at the robot completely for a prolonged period of time). In such cases the robot will explicitly ask the interaction partners if they are interested in continuing the interaction. The answer to this question (or reaching an optional timeout) will determine the result of this activity.

4 Function Analysis

This section describes which functional modules are required to realize the scenario. We explain what they do from a functional point of view, including methods, related work and role for the scenario. Each module has a responsible partner identified. In the case there are more partners involved in one module, their particular roles are defined.

We group the modules into four categories, for each of them listing included modules in a separate subsection. Functions related to video and images processing are described in Subsection 4.1. This category consists of tasks like face detection, identity recognition, head tracking, head pose estimation or robot localization. They serve to detect and track attributes of people in the scene and the position of the robot using video cues. Functions performing detection based on audio cues are given in Subsection 4.2. They include sound source separation and localization, sound recognition or speech recognition. Functions in Subsections 4.3 are responsible for multi-modal fusion of outputs provided by the vision and audition modules at different levels of abstraction. Finally, Subsection 4.4 covers modules dedicated to the implementation and organization of the robots behavior. They include a high-level coordination unit which initiates and plans the overall actions of the robot performed in the scenario.

Please note, that the component-based and event-driven software architecture used for the integration of the modules introduced subsequently will be explained in Section 5.

4.1 Vision

4.1.1 Face Detection

The face detection module is important as a base for several other modules that perform detection/recognition based on facial features. We will use an algorithm we developed within another project [23]. It detects faces that are near-frontal (yaw in the range of ±30 degree). No algorithmic changes are planned in this case.
Responsible partner: CTU

4.1.2 Identity Recognition

This functionality will allow the robot to assign the face tracks to a set of pre-learned identities. The robot can learn a new identity from a short video sequence of a given person which is obtained, e.g., when the person is introduced to the robot. Furthermore, the robot is able to match tracks of unknown identities which have been seen in different times. For example (cf. *Search for Interaction Partner* activity, Section 3.2.6), it allows to recognize that Nao is seeing a person it has already met. In addition, the functionality is needed to maintain an

eye-contact with a person during interaction since the person's face can occasionally disappear, e.g. when the person quickly turns his/her head or steps aside which often happens in real communication. The implementation will utilize ideas presented in [4] and [14].
Responsible partner: CTU

4.1.3 People Categorization

This module classifies people into generic categories based on their face tracks. For example, the robot can categorize people according to their gender (male/female), age groups (baby/young/adult/senior), facial expression (smiling/not-smiling), eye wear (none/eye-glasses/sunglasses) etc. For the year 2 demonstrator, we will concentrate mainly on gender and age categorization which are probably the most useful for robot-human interaction. The categorization of people into generic categories helps the robot to obtain additional useful information besides the identities. For example, the gender is useful to correctly address a person during interaction (*Interaction Opening* activity, Section 3.2.2). As for the use of the age categorization, the robot can use different language expressions when speaking to young people and when speaking to seniors. The algorithm is based on our research on Linear SVMs [24].
Responsible partner: CTU

4.1.4 Visual Speaker Detection

The robot can recognize whether a person whose face is visible is speaking or not. The recognition is based on a detection and motion analysis of the mouth region extracted from face tracks and is inspired by work of Everingham et al. [7]. This vision-based speaker detector complements the audio-based one (cf. work of INRIA) which may not be reliable when the robot is speaking as the audio input is deafen by the robot's speaker. Thus the vision-based speaker detector is essential, for example, when a person wants to interrupt the speaking robot by starting to speak (e.g. saying "good bye" will completely interrupt the interaction, as a possible year 3 extensions for the *Check Listener* activity in Section 3.2.7).

The first prototype of this system working off-line will be available at the end of year 2. The fully functional version is planed for year 3.
Responsible partner: CTU

4.1.5 Simple Gesture Recognition/Detection

The algorithm of this module is able to continuously detect and recognize simple gestures/actions, which are learned off-line before. For the scenario, the robot should be able to detect at least a single gesture so that people can interrupt it non-verbally. When Nao is speaking and explaining the paintings, somebody from the audience can lift his or her hand to signalize a question for instance (relates to *Explain Painting* activity in Section 3.2.4).

Currently, the algorithm is still under development. It has not been tested on Nao yet, since the module needs a stereo vision setup. For the year 2 demonstrator, we will be able to show an off-line demo using the POPEYE [28] cameras. We will try to keep the algorithmic complexity as low as possible for a possible online application.
Responsible partner: INRIA

4.1.6 Head Gesture Recognition

We will develop a head gesture recognition module. The main target will be to recognize subtle head nodding useful for identifying visual backchannel and visual inputs to yes/no questions. It will rely on a parametric optical flow estimation followed by a frequency decomposition module. This generic backchannel will be integrated in the multi-modal dialog module and contributes as such to many dialog situations, e.g., the *Quiz* activity described in Section 3.2.5.
Responsible partner: IDIAP

4.1.7 Head Tracking, Head Pose Estimation

The robot detects and tracks heads of multiple persons in the scene. The tracking is not sensitive to head poses. For every tracked person the head rotation angles (pan, tilt, roll) are estimated. The output of this module is important for the visual focus of attention estimation (4.3.1).
Responsible partner: IDIAP

4.1.8 3D Localization of Faces/Heads

The goal is to place a detected face in the 3D space. This information will help to determine which people are close to the robot and thus are potentially interested in an interaction (see *Approach Human* activity in Section 3.2.8).

For each of the faces detected in one image, the module computes:

- The matching face in the other image (if any). This is done by searching for a similar face description near the epipolar line corresponding to the detected face.
- For those face seen in both images, compute the 3D triangulation. The well-know DLT procedure will be applied to reconstruct the 3D position [10].

INRIA is in charge of providing the two functions and the module. IDIAP will cooperate with INRIA as well to use 3D localization in conjunction with head tracking. CTU is a collaborator since they provide the face detector module.
Responsible partners: INRIA, IDIAP, CTU

4.1.9 Robot Localization

The robot is able to recover his pose (position and orientation) in a previously explored room. Furthermore, knowing the positions of individual paintings in the room, the robot can find the closest piece of art which allows to select the appropriate set of explanations (*Explain Painting* activity, Section 3.2.4). The functionality will be achieved by extending the localization scripts developed during year 1, based on a state-of-the-art structure from motion method [26], and will make use of the upgraded version of the robot's head (new cameras). When put into an unknown room with paintings on the walls, the robot will first walk around and build a sparse 3D model of the place. The walking trajectory will be iteratively corrected using the model just being built. Finally, the accuracy and consistence of the model will be improved by a loop

closing technique. The labels of the individual paintings will be manually input into the model for the year 2 demonstrator, automatic or assisted label assignment is planned for year 3.
Responsible partner: CTU

4.2 Audition

4.2.1 Sound Recognition

This module finds segments of interest in the input audio stream and labels them with the name of a category. It requires a preliminary training phase where all the sounds from a previously recorded database are represented as sparse vectors using supervised learning. The classification of a new unknown sound is a problem of finding the best cluster to its sparse vector representation. This module is derived from [17]. We did some improvements to adapt this algorithm to a real time robotic situation. We will build, with Nao, a database of realistic sounds to test the performances (door, foot steps, keyboard, body sounds, can opening, etc.). In the scenario, recognized and localized sounds will help to track people. For example, a person can be detected based on foot-steps, as described in the *Search for Interaction Partner* activity (3.2.6).
Responsible partner: INRIA

4.2.2 Sound Source Separation and Localization

Multiple sound sources could be playing simultaneously in the scene, e.g., people talking, foot steps, music (year 3). This mixture is recorded by (at least) 2 microphones. Our algorithm is able to statistically separate and locate the sound sources through an iterative procedure, such that it produces signals which are similar to the emitted ones, together with their directions (azimuth, elevation). This procedure requires a preliminary training phase were the robot records white noise from various motor states in order to learn its Interaural Transfer Function.

This functionality will be used as a preprocessing step to enhance the audio stream before the automatic speech recognition module (4.2.5) is applied. The number of sources is known a-priori (has to be given by other modules). The algorithm performance can be improved if it is initialized with an estimation of the sound sources' directions. This could be given by 3D faces localization (4.1.8), considering only speaking faces (4.1.4). The algorithm is still under development. We did not test it on Nao's microphones. In year 2 we will only show an off-line demo using the POPEYE microphones.
Responsible partner: INRIA

4.2.3 Audio Cue Extraction

The module extracts audio cues from the raw stereo audio signal. A good example is the Interaural Time Difference (ITD). State-of-the-art methods based on cross-correlation will be used to compute the ITD values. INRIA is primarily responsible for this module, however, there will be a strong collaboration with ALD since they already have experience implementing audio cue extractors on Nao.
Responsible partners: INRIA, ALD

4.2.4 Association of Audio Cues with 3D Locations

A direction of the sound source can be estimated by several cues from audio-signals recorded by spatially displaced microphones such as the four microphones on the Nao's head. The system has to be properly calibrated. Then we can statistically associate the detected speaking 3D face (4.1.4, 4.1.8) with the audio signal. This will help the robot to localize a speaker during dialogs and turn the head in the right direction when responding. It is useful e.g. for *Quiz* activity (3.2.5).
Responsible partner: INRIA

4.2.5 Speech Recognition

Based on a predefined grammar or a trained speech model this module will recognize spoken sentences. It is required, because some of the robot's activities need to analyze verbal feedback (3.2.2, 3.2.7, 3.2.5). The software [9] was developed in a different project and will only be maintained. No algorithmic changes are planned.
Responsible partner: BIU

4.3 Multi-sensor Fusion and Integration

4.3.1 Visual Focus of Attention Estimation

This module works in conjunction with the face tracking and head pose estimation (4.1.7) and provides inputs to engagement tracking (4.3.3). For a tracked person it is able to tell, whether there is a particular target of visual attention and what is it. The target can be another tracked person or an object whose 3D scene coordinates are known. This functionality is important for *Interaction Opening* and *Check Listener* activities.
Responsible partner: IDIAP

4.3.2 Person Tracking

This module will collect all generated cues about persons that the robot perceives and maintain them in a consistent way such that several hypotheses about persons in the scene are created. The hypotheses contain the 3D location of the person as well as other attributes like the speaking state or the VFOA of persons. The tracking is explicitly responsible for maintaining the hypotheses in cases where persons are currently not perceived by the robot's sensors and needs to contain appropriate strategies to either continue keeping up the hypotheses by prediction or removing them if uncertainty is too high. Essentially, this module performs the fusion (as defined in [15]) of all person-relevant information. In order for this module to function with maximal robustness it will need to make a principled approach to deal with complementary (i.e. in cases where cues are missing) and redundant (i.e. when all cues are existent) modality information. Person tracking plays an important role in several robot's activities (see 3.2.6, 3.2.2, 3.2.7 and 3.2.8).
Responsible partners: BIU, IDIAP

4.3.3 Engagement Tracking

A module that integrates visual (and possibly auditory) cues from the other modules and uses these to determine and track the engagement intention (i.e. the intention to interact with the robot, see *Interaction Opening* activity, Section 3.2.2) and engagement actions of the users (*Check Listener* activity, Section 3.2.7), cf. [12].
Responsible partner: BIU

4.3.4 Multi-party Dialog Management

A module that can manage a mixed-initiative dialog (based on [20]) between multiple humans and the robot using multi-modal input and output cues (3.2.2, 3.2.4, 3.2.5). For instance, it issues robot movements commands like turning the head and making explanatory gestures in addition to pure speech generation. In our scenario it is closely coupled with the engagement tracking in order to maintain a model of whom the robot is currently interacting with. Since it is also able to handle multiple intertwined interactions with different persons/groups at a time it also has to handle the different interaction histories and states.
Responsible partner: BIU

4.4 Robot Behavior and Coordination

4.4.1 Task-based Behavior Abstraction

In this module, we will further develop and integrate behavior generating functions relevant for the activities outlined in the sub-scenarios in Sections 3.2.1–3.2.5 such as a synchronous speech and gesture generation components for Nao based on an implementation of the Task-State Pattern [16]. Applying this pattern to the design of robot software components establishes a common task-based interface and an explicit life-cycle model for task requests eventually simplifying the development of coordination components.
Responsible partner: BIU

4.4.2 Navigation and Motion Control

The paintings in the scene are located at different places. The robot must be able to walk from one painting to another (see Sections 3.2.1 and 3.2.4). It is also necessary to approach visitors (3.2.6, 3.2.8). A navigation module is required to fulfill these tasks. There will be two navigation solutions for the year 2 demonstrator. CTU will provide navigation based on sparse 3D point cloud reconstruction (see Robot Localization module, 4.1.9) allowing for feedback control to make trajectory corrections. This is important since when instructed to walk straight, Nao usually tends to turn its body and leave the prescribed trajectory. A risky part of the outlined solution could be the algorithmic runtime complexity. This is the reason why ALD will prepare a baseline implementation of a navigation algorithm which uses visual landmarks placed in the scene. The robot will simply navigate to marks placed at important positions. Both algorithms will be encapsulated as task-based behaviors using the exact same interface such that a later exchange of the navigation component is transparent from an API perspective.
Responsible partners: CTU, ALD

4.4.3 High-level Behavior Coordination

This module is the top level control unit responsible for the initiation and planning of the robot's overall actions performed in the scenario setup. In a robotics context, this module resembles the role of an executive [13] as a coordination engine for the robot's overall behavior. While on the one hand this module organizes the sequencing of the high-level activities, it additionally needs to to consider conflicts on the level of individual task requests issued as part of autonomous processing in other modules or as a result of mixed-initiative interaction. We plan to employ formal coordination models for this task, which will be rather standard, but a particular focus will be set on an arbitration scheme that specifically supports perception and interaction tasks, e.g., by contextually reducing the self-induced noise from motors or cooling fans to improve the quality of audio signals.
Responsible partner: BIU, ALD

5 Software Architecture

The functional modules defined in Section 4 need to be implemented in software to actually realize the intended scenario on the robot. This process necessitates the definition of several guiding principles to achieve a clearly structured software architecture which accounts e.g. for

- a large number of heterogeneous functional building blocks that form the overall system
- a distributed system with multiple nodes to provide sufficient computational power for AV algorithms
- heterogeneous nodes in this system (e.g. Nao vs. usual desktop computers)
- a distributed scientific development process

The consortium will approach these challenges by applying techniques of component-based decomposition and Event-based Systems [8]. Each of the functional modules described before will be realized by one or more software components, depending on the coherence of the functionality or reusability of processing results for other components. Responsibilities for the components are given according to the partners assigned for each functional module. In the following, functional modules defined in Section 4 will be termed *modules* and their actual realization as a software component will be termed *component*.

5.1 Communication Middleware and Message Formats

Defined software components are not independent and need to communicate in order to exchange information. For this purpose we will continue to use the Robotics Service Bus (RSB) middleware [27] that provides a hierarchical message bus across heterogeneous nodes. Its event-based design allows to reduce the coupling between components in large distributed systems like the robotic system envisioned for this demonstrator (cf. [8, chapter 1]). RSB provides implementations for C++, Java, Python, Common LISP and Matlab (via Java) and hence accounts for the integration of components implemented in different languages, which is also necessary to integrate existing software. Moreover, it has a lightweight footprint which enables it to operate on Nao.

Besides the middleware, common message formats are required to establish communication between components. After having evaluated several alternatives (LCM [11], MessagePack [1], and ROS MSG [2]), we decided to utilize Google Protocol Buffers [3] as a representation and serialization format for messages and to maintain a repository of types defined by their IDL specification. Protocol Buffers integrate well with the chosen principles by providing type-safe access to data in all required programming languages while still allowing for component evolution with backward-compatible deserialization mechanisms. The second aspect is especially important for the distributed integration where components are continually modified by different people. The centrally maintained repository of IDL specifications aims at avoiding duplication of representations for comparable tasks and message contents. Listings 1 and 2 show two examples of IDL specifications that will be used.

The application of RSB and Protocol Buffers will form a basis for the scientific work with Nao, with enhanced features compared to Nao's own middleware NaoQi. Nevertheless, we will not replace NaoQi and reimplement functionality for the control of the robot provided by NaoQi. Instead, automatic translation technologies between RSB and NaoQi will be evaluated and applied to provide a RSB wrapper around NaoQi's functionalities. This will e.g. make all sensory results of Nao available on the RSB message bus.

5.2 Component Decomposition

Software components that realize the modules described before can be categorized into several types as depicted in Figure 6. Sensing modules provide sensory information gathered from Nao to other components of the system, while actuation components provide interfaces to control the robot. Sensory information can be both information that is gathered about the environment using the robot's sensors as well as proprioceptive information like the joint angles that will also be made available to other parts of the system. The majority of vision and audition modules described in Section 4 extract cues about people around the robot. They are grouped under the

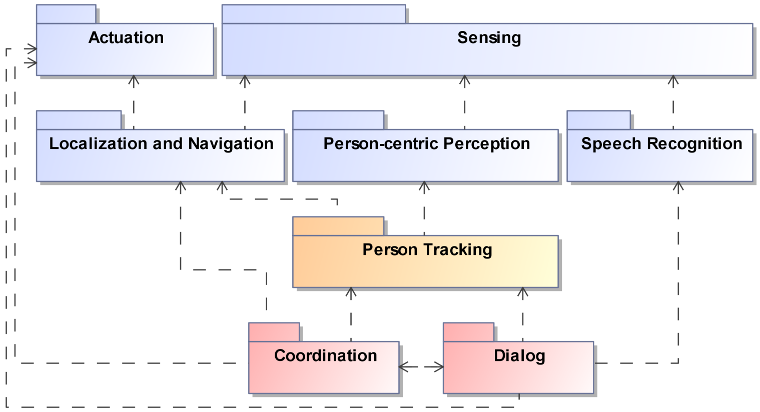


Figure 6: Schematic overview of software component types and their dependencies.

Listing 1: An exemplary Protocol Buffers-based IDL for monocular images

```
message ImageMessage {

    enum Depth {
        DEPTH_8U = 8;
        DEPTH_16U = 16;
    }

    enum ColorMode {
        COLOR_GRAYSCALE = 0;
        COLOR_RGB = 1;
        COLOR_BGR = 2;
        COLOR_YUV = 4;
        COLOR_YUV422 = 8;
    }

    enum DataOrder {
        // separate planes
        DATA_SEPARATE = 0;
        // interleaved image
        DATA_INTERLEAVED = 1;
    }

    required bytes data = 1;
    required uint32 width = 2;
    required uint32 height = 3;
    optional uint32 channels = 4 [default = 3];
    optional Depth depth = 5 [default = DEPTH_8U];
    optional ColorMode colorMode = 6 [default = COLOR_RGB];
    optional DataOrder dataOrder = 7 [default = DATA_SEPARATE];

}
```

term “Person-centric Perception”. By processing the audio and video channel they provide a collection of cues that still needs to be assembled into person hypotheses, a high-level construct that provides a combined view on all available cues of currently perceived people in the scene in a coherent and consistent manner. This is an essential requirement to reduce the complexity of dialog and control components and the task to maintain these hypotheses will be carried out by a dedicated category of components (“Person Tracking”). Dialog and Coordination components will control the high-level behavior of the robot, especially the interaction with people. This also requires the location of the robot on the table, which is provided by another type of components. Navigation, as well as the coordination actually control the robot and hence require access to the actuation components, resulting in a need of coordination strategies.

The remainder of this section will highlight key aspects of how functional modules will be decomposed into software components of the aforementioned types and their integration into architectural concepts. UML2 component diagrams are used for the description. While they provide a simple overview about the involved components and their communication paths, they lack many important aspects for a full system description. These include:

- Timing aspects and guarantees
- Resource allocation and consumption
- Further non-functional properties

To further specify these aspects we evaluate the use of advanced modeling languages like MARTE [18], a UML2 profile for real-time embedded systems.

Listing 2: An exemplary Protocol Buffers-based IDL for results of a component that calculates the VFOA of visible faces in the scene

```
message VfoaMessage {

    required uint32 imageWidth = 1;
    required uint32 imageHeight = 2;

    message BoundingBox {
        required uint32 x = 1;
        required uint32 y = 2;
        required uint32 width = 3;
        required uint32 height = 4;
    }

    message Target {
        required string name = 1;
        required float probability = 2;
    }

    message Person {
        required uint32 id = 1;
        required BoundingBox box = 2;
        repeated Target targets = 3;
    }

    repeated Person persons = 3;

}
```

5.3 Robot Control and Coordination

Controlling the robot as an actuator requires coordination of different, potentially conflicting, commands. We will base our coordination scheme on a control interface for Nao based on the Task-State Coordination Pattern [16]. Therefore, a set of task-servers will be implemented that wrap or combine several NaoQi functions to create meaningful tasks units. These task-servers will form the only means of controlling the robot. The well-defined task definitions will then be used to evaluate coordination schemes with either a central coordination component or in a distributed fashion. In any case, coordination needs to respect the strong focus of this project on sensing, which e.g. means that the robot may have to stop certain motors for auditory tasks or needs to reduce camera motion for vision applications.

To fulfill the scenario, at least the following task-servers are planned realizing the *Task-based Behavior Abstraction* module as components of the software architecture:

TTS This task-server will control the text-to-speech engine of Nao and provide intermediate results about spoken words.

Locomotion A task-server that controls the locomotion abilities of Nao to navigate on the table.

Gesture A task-server that initiates gestures of the robot like pointing at paintings for the Explain Painting activity (see Section 3.2.4).

Depending on the granularity of certain tasks like a visual search for people in the scene more task-servers will be added. In general, task-servers encapsulate coherent *behaviors* of the robot and more complex behaviors may use several lower-level behaviors (implemented as task-servers). A behavior is therefore a connection of sensory inputs with actuation capabilities on a lower level than the depicted high-level control scheme in Figure 6 where the robot is controlled only be the coordination components. Moreover, every behavior is controlled using the Task-State pattern. This allows a tighter coupling of sensors and actuators with the advantages of reduced complexity in the coordination components and better support for timing requirements. An example for a behavior utilizing other behaviors is navigation, which combines sensory processing with controlling the robot's movements using the locomotion task-sever. Summing up, this creates a hierarchical structure which needs to be coordinated and where appropriate mechanisms will be evaluated. Please note, that the term behavior does not imply a fully behavior-based robotic architecture [5] in our system.

From a technical perspective NaoQi already provides possibilities to define flat behaviors using the Behavior Manager. A good integration in Choregraphe [22] especially to design motions is available. For basic behaviors (especially motion) the task-based system will be built on the NaoQi behaviors and enhance them with the more powerful protocol and integration in the coordination scheme.

5.4 Synchronization and Memory

We will further rely on memory as a part of the architecture, but in a more distributed and middleware-integrated version than the presented m^3s system [27, previously called EgoMemory]. The concepts of m^3s will be converted to specialized components that directly integrate

memory processes	⇒	regular (RSB) components
memory layer	⇒	temporal (RSB) buffer
memory IDs	⇒	uniform resource identifier's
memory subscription	⇒	buffer subscription
timing / lifeline operators	⇒	predictor (RSB) components
representation format	⇒	IDL-specified binary format (Protocol Buffers)
representation content	⇒	flexible, as needed / agreed
content-based query	⇒	XPath-based query on binary format
egocentric memory	⇒	buffer with egocentric representation

Table 1: Transition from m^3s memory concepts to RSB-based distributed memories.

into the RSB middleware and rely on the Protocol Buffer representation format. As a core, layers will be converted into buffers, which subscribe on RSB events and provide a short-term history of these events. These buffers will expose an RPC interface to access the contents with queries and provide event-based access to temporal changes of contents as already planned for the m^3s system. To make contents of buffers addressable, RSB will be enhanced with an URI scheme for events. The prediction abilities of m^3s will be converted into special components that utilize normal RSB mechanisms. All these modification aim to make the memory more flexible and generally usable, on the one hand by decomposing previously assembled functionality into distinct components, and on the other hand by focusing on a single representation format. Flexible access to these Protocol Buffers-based representations will be given by providing an implementation the XPath [6] specifications. Table 1 gives an overview of how concepts from m^3s will map to concepts of the new memory scheme.

The memory abilities of the architecture will be used differently depending on the architectural level. On the lower levels they will most likely be used for synchronizing processing results of different components, where computation have different processing times. For higher-level abilities like the dialog or engagement calculations the temporal subscriptions provide discretized triggers, e.g. when a person approaches the robot. Moreover, a stabilization of processing results with temporal knowledge is intended. Another application is providing short-term knowledge about the scene the robot acts in. The dialog can answer questions or adopt its behavior by querying knowledge from these buffers.

5.5 Person-centric Perception Components

The extraction of cues about persons in the scene forms the major amount of functional modules described in Section 4. Starting with the vision-only processing pipeline, this section describes planned components of the overall software architecture that form this functional block. Figure 7 illustrates a first iteration of the components for vision-only cue extraction starting with the streamed video from Nao's camera and ending with a dedicated component that joins the person cues into consistent person hypotheses. Components that extract cues only publish the generate cues and not the image these cues were based on for saving bandwidth. Hence, further processing on the generated results needs to associate these results with the original image they were based on. The aforementioned buffers will provide the basis for this synchronization

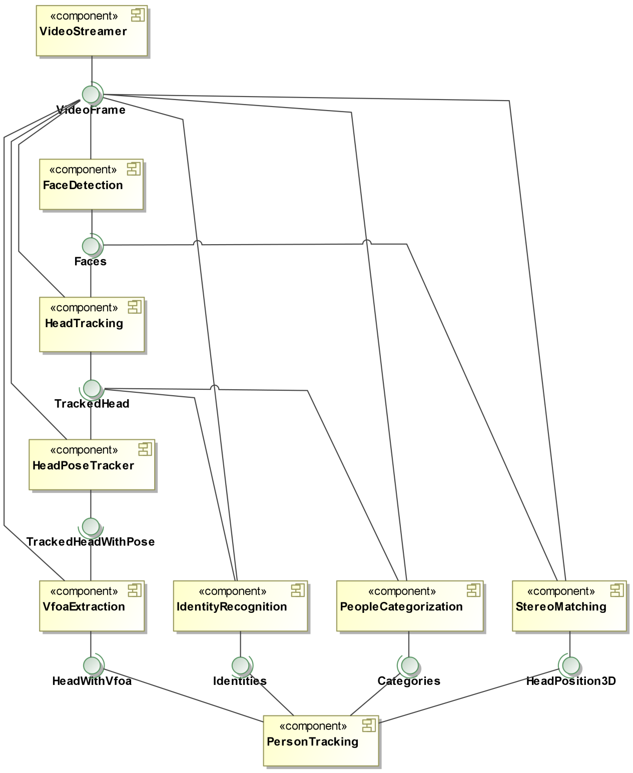


Figure 7: Initial draft for vision-only components involved in extracting cues about persons. Information flow is event-based from top to bottom.

process and therefore are installed at required places. In the diagram all exchanged data needs to be buffered to create a synchronized view on all cues for the PersonTracking component.

Figure 8 displays the initially planned components that process auditory information to extract information about persons in the scene. Again, synchronization elements are likely to be necessary in the connection paths to ensure consistent hypotheses. Please note the special role of the “AVCueAssociation” component which performs a low-level fusion of visual and and auditory information to associate produced sounds with the generated 3D locations of faces. It realizes the functional module *Association of Audio Cues with 3D Locations*. The generated sounds and speech hypotheses are not directly used by the person tracking. Instead they will be utilized by the higher-level dialog and behavior generation components.

5.6 Deployment Strategy

The implemented components need to be deployed on either Nao or one of the external computers in a way that respects requirements like the limited processing prower of Nao and bandwidth or timing requirements. Moreover, the complexity of dealing with two middlewares needs to be reduced. Hence, we will start with a deployment scheme as depicted in Figure 9, where the adaption of Nao’s functionality will be performed directly on the robot, resulting in all external communication with the robot being performed using RSB. All other processing will

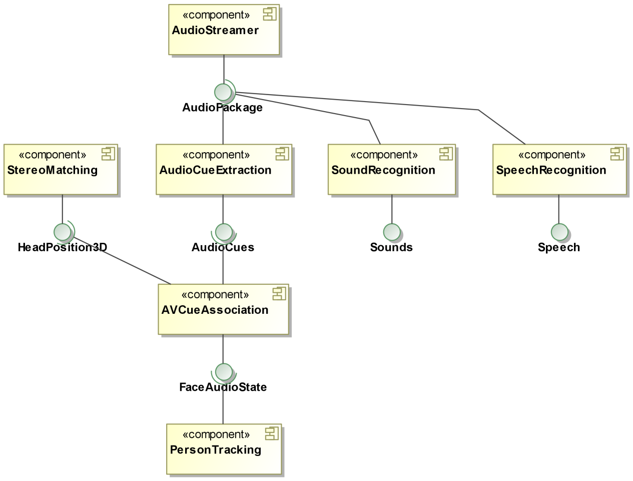


Figure 8: Initial draft of auditory components that extract cues about persons.

be performed on external computers. This frees the limited CPU of Nao from high computational load by scientific software and thereby allows the functionality integrated in NaoQi to work with enough performance for time-critical robot control. On the other hand, this requires constant transmission of audio and video over the network. With the availability of the new head for Nao (see Phase 3 – Basic Interactive Setup on HUMAVIPS Nao Head in Section 7.3), which has increased processing power, we will evaluate the option to deploy selected components on the robot, but still using the RSB interfaces for Nao. Either the loopback device or an efficient in-process communication are available through RSB. Even though, from the middleware-perspective modifications to the components are minimal for deploying them to the robot, a cross-compilation is required to operate on Nao, graphical debugging is cumbersome and the set of available libraries is limited. Therefore, we will use this option only if timing requirements cannot be met otherwise.

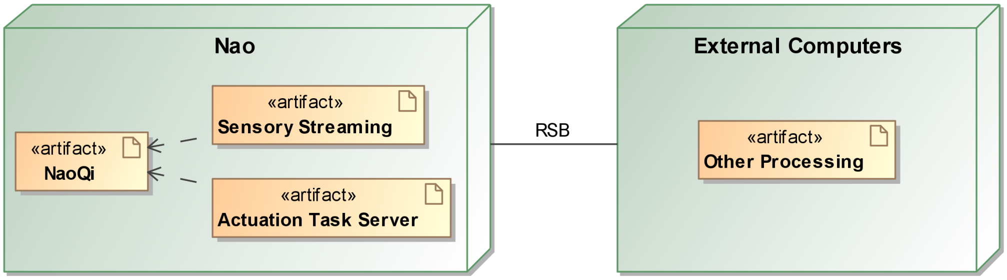


Figure 9: Deployment scheme for the demonstrator.

6 Evaluation Strategies

Evaluation is a crucial process to measure the progress of the project and to compare the performance of individual algorithms and the integrated system to other scientific and industrial efforts. Furthermore, it is essential for the project partners themselves to continuously conduct evaluation steps (synchronized with the phases cf. 7) in order to monitor our progress in year 2 and detect regressions. Moreover, a dedicated phase at the end of year 2 is defined to perform a final evaluation of this year's achievements, which takes into account the continuously generated evaluation results.

To ease evaluation and development, it will be beneficial to perform certain evaluation steps only on parts of the overall scenario in addition to the evaluation of the overall system. The rationale behind this is that the scientific expertise of the respective partners is essential to perform a through evaluation of a specific scientific contribution. Furthermore, even an evaluation of single activities or sub-scenarios as defined in Section 3 can yield relevant results. Please note, that in contrast to year 1 we aim to perform these evaluation steps within the developed software architecture, either in defined smaller experiments with the developed system or on the basis of replayed data from the dataset to be recorded in year 2 (which will be recorded in the same setting as the year 2 scenario).

Evaluation can be performed at different levels in a complex robotics system and with different perspectives. For year 2, we want to perform evaluation and benchmarking at the *platform*, *framework*, *component*, and *system* levels, which will be briefly explained in the remainder of this section.

6.1 Platform-level

An important share of research and development efforts in HUMAVIPS are devoted to the development of a new "head" for the humanoid robot NAO. This is not only relevant for the project due to improved sensing capabilities but also due to a much more powerful embedded PC board allowing more complex onboard processing. Thus, the platform-oriented evaluation steps in year 2 will focus not only on the new sensors but also on the implications of the new embedded PC on software architecture and deployment strategies.

With a first prototype of the new HUMAVIPS head for Nao available, an initial evaluation will be carried out to verify the applicability of the chosen design for the project aims. This will be already done on representative algorithms also used in the final year 2 system. With the final head and its control software in place, further aspects of the technical realization for the new head design will be evaluated such as the optimal mode and resynchronization frequency for the stereo camera subsystem, performance and reliability of audio and video transmission as well as new control strategies for the fan speed and its impact on audio signal quality.

6.2 Framework-level

This perspective will focus on the software architecture of the whole system as well as the integration procedures. In order to evaluate these perspectives, we will perform a quantitative evaluation of the middleware approach used in HUMAVIPS on the embedded PC of the NAO as well as on dedicated workstations and compare it to other state-of-the-art frameworks such as NaoQi's SOAP-based middleware stack, the ROS middleware core and YARP.

To assess the developed software architecture we aim to employ common software and development metrics for this purpose (e.g. cyclomatic complexity, code rank, ...) at the level of individual components as well as on the overall architecture and qualitative assessments using techniques from empirical software engineering. Last but not least, an important evaluation feedback will be provided in year 2 through the HUMAVIPS partners as the primary users of the framework and the integration concepts.

6.3 Component-level

On this level, the performance of the individual modules, cf. Section 4 will be analyzed in terms of both their accuracy and their processing characteristics. To evaluate accuracy, several sources of annotated data, e.g., the year 2 HUMAVIPS dataset will be considered. To provide reasonable test data, the year 2 dataset will feature situations of different complexity levels recorded and annotated with ground-truth data in the sub-scenarios introduced in Section 3. For instance, for assessing the person perception modules, different complexity levels could be defined as follows:

- Nao is not moving; one person moves in its field of view (FOV), looks in several directions, speaks and addresses Nao, and react to Nao's speech;
- Nao is allowed to move its head; the tracked person can disappear from the FOV, and speak will not being seen;
- Nao is not moving; two person's interact with each other and addresses Nao from time to time;
- The same as before, but Nao is authorized to nod, or/and move its head to look directly at each person.

Please note, that the detailed specifications of these situations are developed as part of the year 2 dataset recording task. Furthermore, as annotation can be tedious and time consuming, and also to allow an early and ongoing comparison with other works in the literature, standard benchmark data will be exploited, including the year 1 data recorded with the POPEYE robot.

Besides accuracy, further evaluation criteria will be defined and applied. For instance, an analysis of the computational profile of the individual algorithms in real working conditions will be relevant for taking informed design decisions at the architectural level. Examples for relevant criteria from this perspective include processing speed (in millisecond/frame for instance), memory usage, the processing variability or latency.

Another, equally important perspective for component-level evaluation is to assess their robustness to varying processing conditions. Most evaluation on benchmark data assume that the data is processed at full frame-rate. In real-world conditions, most vision and audio modules do not process the data at this rate. Evaluation of the accuracy of the algorithms at their expected processing speed (which may include variability in the sampling rate) will be conducted through frame acquisition perturbation simulations.

6.4 System-level

At this level we will take on a functional perspective evaluating the overall performance of the complete system in the given scenario. For this purpose, appropriate metrics and qualitative test cases will be defined and maintained along with the incremental development strategy. These metrics and test cases will capture especially interactional success, e.g., overall task completion (e.g., explanation of all paintings to a single user) or just the durations of an engaged interaction, through initial user studies. Moreover, the quality of certain complex activities such as the robots positioning towards humans or the understandability of pointing gestures (e.g., to paintings) ar e functional properties that can be evaluated. Please note, that the interactional success also depends on the reaction time and perception performance of the overall integrated system which is expected to outperform the performance of the individual algorithms. As a major hypothesis of the project, this aspects will also be considered and evaluated at this level although similar methods as on the component level will be applied.

7 Roadmap

To establish an efficient system integration process towards the year 2 scenario, the HUMAVIPS partners will implement an incremental development strategy with continuous integration and testing procedures at every partner's site. In comparison to year 1, all partners committed to realize and replicate the same scenario locally such that a decentralized and ongoing development of components can be performed. Furthermore, the recording of the year 2 dataset will be carried out in the same scenario and environment. This and dedicated support for using the offline dataset in the HUMAVIPS software architecture will allow experimentation on relevant data using the same components as developed for the scenario implementation.

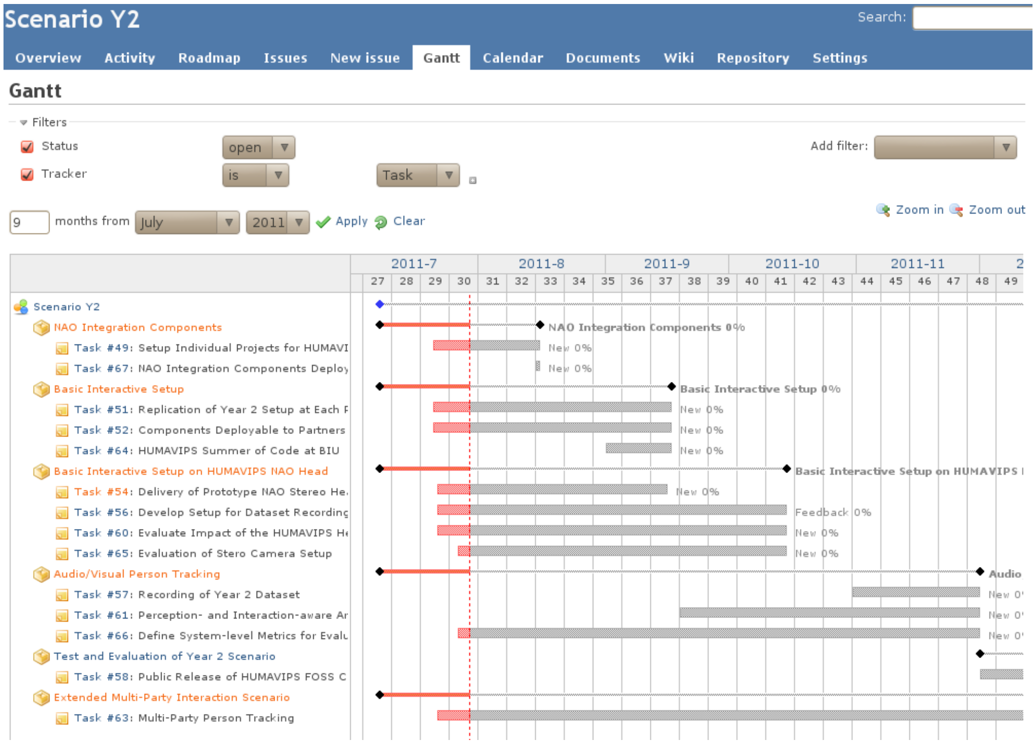
To organize our efforts from a project management perspective, the HUMAVIPS partners agreed on a roadmap with several phases of increasing complexity and specific focus. Each phases ends with a dedicated milestone where a specific set of functionality has to be achieved and delivered to all partners to stay in sync with the decentralized development.

In the remainder of this Section the different phases and their partially interleaved scheduling will be briefly described. The details of these phases are available in a dedicated year 2 project[1] in the HUMAVIPS collaboration environment where also the implementation progress is continuously monitored from the WP7 project management team and replanning is carried out if required. Figure 10 shows a screenshot of the planning part of that environment. Specific responsibilities of individual partners are also assigned at the fine-grained level of tasks and issues defined in that project.

7.1 Phase 1 – Nao Integration Components

Subsequent to project meetings in May and July 2011, all partners agreed that a promising way to achieve integration will be to replicate the fundamentally same scenario at each partners site. Due to the shared availability of the Nao robotics hardware, this is easily possible. However,

[1] https://code.humavips.eu/projects/scenarioy2, login credentials are available upon request



☑ Status open

☑ Tracker is Task

9 months from July 2011

Scenario Y2	
NAO Integration Components	NAO Integration Components 0%
Task #49: Setup Individual Projects for HUMAVI	New 0%
Task #67: NAO Integration Components Deploy	New 0%
Basic Interactive Setup	Basic Interactive Setup 0%
Task #51: Replication of Year 2 Setup at Each F	New 0%
Task #52: Components Deployable to Partners	New 0%
Task #64: HUMAVIPS Summer of Code at BIU	New 0%
Basic Interactive Setup on HUMAVIPS NAO Head	Basic Interactive Setup on HUMAVIPS I
Task #54: Delivery of Prototype NAO Stereo He.	New 0%
Task #56: Develop Setup for Dataset Recording	Feedback 0%
Task #60: Evaluate Impact of the HUMAVIPS He	New 0%
Task #65: Evaluation of Stero Camera Setup	New 0%
Audio/Visual Person Tracking	Audio
Task #57: Recording of Year 2 Dataset	New 0'
Task #61: Perception- and Interaction-aware Ar	New 0'
Task #66: Define System-level Metrics for Evalu	New 0'
Test and Evaluation of Year 2 Scenario	
Task #58: Public Release of HUMAVIPS FOSS C	
Extended Multi-Party Interaction Scenario	
Task #63: Multi-Party Person Tracking	

Figure 10: An exemplary subset of issues from the year 2 project in the HUMAVIPS collaboration environment that are categorized as tasks (14 of currently 46 tracked issues).

to conduct development and experimentation in the same environment, not only the hardware but also the physical environment, cf. Section 2 as well as the complete software architecture with all relevant components will be set up and used by each partner.

To allow easier scientific experimentation with the Nao humanoid robot and integration of components in a coherent and sustainable software architecture, the robotics middleware provided by the partners from BIU will be used in combination with the robot control API provided by ALD. Phase 1 focuses on further integration and deployment of the Nao-related components in the target middleware as well as their deployment to project partners as a first step towards setting up a replicated development and experimentation environment for the year 2 scenario.

Rationale Foundation for A/V experiments with Nao and year 2 component development.

Schedule 15.07.2011 – 15.08.2011

Implementation Major development and deployment tasks in this phase are:

- Streaming components for NaoQi's timestamped A/V data.
- Streaming components for Nao's joint values.
- Recording and replaying tools for A/V and joint data.
- Setup of portal projects for anticipated year 2 software components by all partners.

- Final agreement on development conventions for HUMAVIPS FOSS components.

Verifiable Goals Required to complete this phase:

1. Initial version of Nao components operational at each partners site.
2. Portal projects for all year 2 components set up and ready for use.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/5`

7.2 Phase 2 – Basic Interactive Setup

While the activities introduced so far primarily consider infrastructure aspects, the activities in phase 2 – carried out in parallel to phase 1 tasks – are focused on the replication of the experimental scenario as described in the environment specification, cf. Section 2, and the scaffolding of the system architecture described in Section 5. This up-front integration step will allow for an early assessment if the envisioned concepts at the architectureal level are suitable within the experimental scope of HUMAVIPS both from an engineering and scientific perspective. Hence, the focus of this phase will be on the functional integration of an initial set of modules required for an implementation of the sub-scenarios[2] outlined in Section 3. While this initial set will allow us to develop a basic scenario with limited complexity, it also guarantees that every partner will already contribute and technically integrate at least one of the functions described in Section 4 in this early stage of year 2 developments. The complexity of the scenario will be limited in this phase for instance by restricting the number of simultaneously interacting persons to a single adult or simple marker-based navigation. Subsequent phases will relax these restrictions and thus increase the complexity of the overall challenge.

Rationale Basic interactive scenario for scaffolding the system architecture.

Schedule 15.07.2011 – 16.09.2011

Implementation Major development and deployment tasks in this phase are:

- Integration of partner components into system architecture.
- Replication of environment at each partners site.
- Evaluation of basic interactive setup at each partners site.
- Development of a controller for semi-autonomous dataset recording in Wizard-of-Oz type studies.

Verifiable Goals Required to complete this phase:

1. Basic interactive setup operational and tested at each partners site.
2. Software components for phase 2 available through Open Portal.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/2`

[2] Except the Quiz activity which will be realized in phase 4.

7.3 Phase 3 – Basic Interactive Setup on HUMAVIPS Nao Head

A fundamental prerequisite for rich visual processing of information from the three-dimensional world is the ability to efficiently process pairs of synchronized images recorded in a stereo camera setup. For this reason, an important milestone in the HUMAVIPS project is the availability and integration of a novel Nao robot head with an improved stereo vision subsystem developed in WP7 by the ALD partner. The availability of this new head is in many ways fundamental as it allows research on novel stereo vision algorithms on a standard robotics platform which in turn will improve multi-modal fusion and provide completely new cues to the higher fusion and decision layers in the robots system architecture. This development will ultimately allow the project partners to implement the behavioral competence we envision to demonstrate at the end of the HUMAVIPS project.

An important milestone towards this goal will be reached in year 2 with the activities carried out in this phase where the new hardware will be integrated and evaluated in the basic interactive scenario as developed in phase 2. This integration and evaluation will be done primarily in close collaboration between ALD and BIU partners. The rationale behind this is to provide the new head already completely integrated in the infrastructure developed as part of phase 1 to the HUMAVIPS partners. Furthermore, BIU will provide test data and updates to the phase 1 toolchain as soon as possible to the partners and also prepare the data recording setup for year 2 which is bound to the availability of the new head.

Rationale Integration and evaluation of new HUMAVIPS stereo head in basic scenario.

Schedule 16.08.2011 – 14.10.2011

Implementation Major development and deployment tasks in this phase are:

- Evaluation of prototype head without NaoQi but with V4L[3] API.
- Extension of A/V streaming, recording and synchronisation toolchain to stereo head.
- Analysis of changes imposed by new head on system level (e.g., faster processor).
- Evaluation of new vision subsystem against baseline hardware (standard Nao head).

Verifiable Goals Required to complete this phase:

1. Basic interactive scenario ported to new HUMAVIPS Nao head.
2. New head deployed to all partners and running in phase 2 scenario.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/3`

7.4 Phase 4 – Audio/Visual Person Tracking

Subsequent to phase 2 and in parallel to phase 3 integration tasks in this phase will be geared towards algorithmic extension and scientific experimentation in a more complex scenario. Research will be focused on further integration and evaluation of A/V fusion methods and improved person tracking as needed for a more natural human-robot interaction. The robot's

[3] Video for Linux

behavior is supposed to be extended by a perception-aware arbitration module taking into account control strategies to optimize the raw sensory input in interactions, e.g., modulating the fan speed if verbal input is expected. Furthermore, the robot's locomotion behaviors will be enhanced, e.g., by improved orientation strategies towards humans in an interaction. Based on the improved gathering of cues about persons, this phase will also contain initial efforts for tracking their engagement.

To evaluate these improvements, the complexity of the basic interactive scenario will be increased. For instance, interacting persons are allowed to turn around during a single interaction, be occluded for a short time without getting lost and longer verbal interactions (also considering mixed initative) will be achieved by realizing the Quiz activity as introduced in Section 3.2.5 in the scenario. Additionally, first tests will be done in a multi-party interaction situation which will also be part of the dataset that is recorded in this phase. All developments in this phase will be targeted towards and eventually tested with the HUMAVIPS stereo head. Furthermore, the HUMAVIPS dataset will be recorded using the software architecture available at this phase and with the stereo head. Within the defined scenario a set of baseline tests and metrics for evaluation of system and component performance will be developed and applied.

Rationale Further integration of A/V and stereo cues and improved person tracking.

Schedule 17.09.2011 – 30.11.2011

Implementation Major development and deployment tasks in this phase are:

- Extended person tracking based on multi-modal sensor fusion.
- 3D localization of persons and objects.
- Improved landmark-based robot localization.
- Basic estimation and tracking of engagement levels of persons.
- Development of baseline tests and metrics for the Year 2 scenario.

Verifiable Goals Required to complete this phase:

1. Utilization of stereo A/V cues can be demonstrated.
2. Baseline tests conducted in the phase 4 scenario.
3. Data recording done with the new stereo head.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/6`

7.5 Phase 5 – Extended Multi-Party Interaction Scenario

The aim of this phase is to further extend the state of the year 2 system such that all elements of the interactive scenario as defined in Section 3 are realized and a stable interaction taking into account the presence and engagement of multiple persons can be demonstrated and evaluated. From a scientific viewpoint, this increased complexity not only requires further work on the individual detection, fusion and recognition algorithms as well as the overall architecture but also requires the adaptation of the engagement modules. Advanced perceptual components such as person recognition and categorization will be added facilitating the robot to display more sophisticated social behavior.

We expect that at this stage, testing of the perceptual subsystems will already be possible both online in the integrated demo but also offline using the same software components as in the online system but with the offline dataset recorded in phase 4. System and component evaluation will be done with the previously developed metrics and performance will be compared against the available baseline tests.

Rationale Develop complex person awareness including engagement models.

Schedule 01.12.2011 – 31.01.2012

Implementation Major development and deployment tasks in this phase are:

- Person tracking extension towards complex multi-party situations.
- Integration of identity recognition and person categorization modules.
- Adaptation and evaluation of engagement modules for groups of persons.
- Further extension of test and evaluation procedures based on the year 2 dataset.

Verifiable Goals Required to complete this phase:

1. Multi-party interaction in the given scenario demonstrated in the system.
2. Test and comparison against baseline system of phase 4.
3. Replicated setup of final system at each site available.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/6`

7.6 Phase 6 – Test and Evaluation of Year 2 Scenario

The final phase towards the year 2 scenario will be dedicated to further testing and evaluation along the technical and functional dimensions briefly outlined in Section 6. Here, the benefits of the de-centralized development and replicated setups will become apparent facilitating parallel and distributed evaluation focused on the aspects important for the individual groups. Algorithmic performance of individual modules can be demonstrated within the scenario either on the basis of data available in the recorded dataset or in scientific experiments using the complete system in a dedicated experiment.

Please note, that activities in this phase start in parallel to phase 4 and are linked with early evaluation of these versions of the integrated system. For instance, the definition of metrics and baseline tests is an activity carried out here, while their application is part of the other phases. This phase ends with the 2nd year review meeting where the scenario will be setup and demonstrated. A further activity in this phase is to disseminate the results of year 2, e.g., in the form of open source software components or open data available to the scientific community via the HUMAVIPS Open Portal website as part of our WP6 activities.

Rationale Testing, evaluation and dissemination of year 2 scenario.

Schedule 01.11.2011 – 29.02.2012

Implementation Major development and deployment tasks in this phase are:

- Definition of baseline tests in the overall scenario and sub-scenarios.
- Definition of metrics for hardware, component and system-level performance.
- Preparation of year 2 dissemination of HUMAVIPS components.

Verifiable Goals Required to complete this phase:

1. Results of quantitative evaluation of year 2 system available.
2. Year 2 HUMAVIPS open source components available in Open Portal.
3. Year 2 HUMAVIPS dataset and toolchain available to the public.

Portal Link `https://code.humavips.eu/projects/scenarioy2/versions/1`

References

[1] Messagepack. `http://msgpack.org/`. visited: 07/27/2011.

[2] msg. `http://www.ros.org/wiki/msg`. visited: 07/27/2011.

[3] Protocol buffers. `http://code.google.com/p/protobuf/`, 2011. visited: 07/25/2011.

[4] N. E. Apostoloff and A. Zisserman. Who are you? – real-time person identification. In *British Machine Vision Conference*, 2007.

[5] Ronald C. Arkin. *Behavior-based robotics*. Intelligent robots and autonomous agents. MIT Press, 1998.

[6] Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernández, Michael Kay, Jonathan Robie, and Jérôme Siméon. *XML Path Language (XPath) - W3C Recommendation 23 January 2007*. W3C, 2.0 edition.

[7] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy” – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*, 2006.

[8] Ted Faison. *Event-Based Programming*. Apress, May 2006.

[9] G. A. Fink. Developing HMM-based Recognizers with ESMERALDA. In Václav Matousek, Pavel Mautner, Jana Ocelíková, and Petr Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin and Heidelberg, 1999. Springer.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[11] Albert S. Huang, Edwin Olson, and D.C. Moore. LCM: Lightweight communications and marshalling. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4057–4062. IEEE, 2010.

[12] David Klotz, Johannes Wienke, Julia Peltason, Britta Wrede, Sebastian Wrede, Vasil Khalidov, and Jean-Marc Odobez. Engagement-based multi-party dialog with a humanoid robot. In *Proceedings of the SIGDIAL 2011 Conference*, pages 341–343, Portland, Oregon, June 2011. Association for Computational Linguistics.

[13] David Kortenkamp and Reid Simmons. *Robotic System Architectures and Programming*, chapter 8, pages 187–206. Springer-Verlag, 2008.

[14] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009.

[15] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems, 1989.

[16] Ingo Lütkebohle, Julia Peltason, Britta Wrede, and Sven Wachsmuth. The Task-State Coordination Pattern, with applications in Human-Robot-Interaction. In Rachid Alami, Rüdiger Dillmann, Thomas C. Henderson, and Alexandra Kirsch, editors, *Learning, Planning and Sharing Robot Knowledge for Human-Robot Interaction*, number 10401 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[17] Richard F. Lyon, Martin Rehn, Samy Bengio, Thomas C. Walters, and Gal Chechik. Sound retrieval and ranking using sparse auditory representations. *Neural Computation*, 22:2390–2416, 2010.

[18] Object Management Group. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*, 1.0 edition, 11 2009.

[19] Object Management Group. *OMG Unified Modeling Language^{TM} (OMG UML), Infrastructure*, 2.3 edition, 05 2010.

[20] Julia Peltason and Britta Wrede. Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns. In *Proceedings of the SIGDIAL 2010 Conference*, Tokyo, Japan, September 2010. Association for Computational Linguistics.

[21] Karola Pitsch, Sebastian Wrede, Jens-Christian Seele, and Luise Süssenbach. Attitude of german museum visitors towards an interactive art guide robot. In *Proceedings of the 6th international conference on Human-robot interaction*, HRI '11, pages 227–228, New York, NY, USA, 2011. ACM.

[22] E Pot, J Monceaux, R Gelin, and B Maisonnier. Choregraphe: a Graphical Tool for Humanoid Robot Programming. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 46–51, 2009.

[23] Jan Šochman and Jiří Matas. Waldboost - learning for time constrained sequential detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 150–157, Los Alamitos, USA, June 2005. IEEE Computer Society.

[24] Sören Sonnenburg and Vojtěch Franc. Coffin: A computational framework for linear svms. In *Proceedings of the 27th Annual International Conference on Machine Learning (ICML 2010)*, pages 999–1006, Madison, USA, June 2010. Omnipress.

[25] C. Szyperski, D. Gruntz, and S. Murer. *Component software: beyond object-oriented programming*. Component software series. Addison-Wesley, 2002.

[26] Akihiko Torii, Michal Havlena, and Tomáš Pajdla. Omnidirectional image stabilization by computing camera trajectory. In Toshikazu Wada, Fay Huang, and Stephen Y. Lin, editors, *PSIVT '09: Advances in Image and Video Technology: Third Pacific Rim Symposium*, volume 5414 of *Lecture Notes in Computer Science*, pages 71–82, Berlin, Germany, January 2009. Springer Verlag.

[27] Johannes Wienke and Sebastian Wrede. Deliverable 2.1: Tutorial on event-driven memory architectures in robotics. Technical report, HUMAVIPS: Humanoids with auditory and visual abilities in populated spaces, 2010.

[28] João Xavier, Rui Caseiro, and Helder Araújo. A human head simulator for research in active vision. available online: `http://miarn.sf.net/pdf/human_head_sim.pdf`.