**Social Robotics Use Cases**

Acronyms:

AMM = Activities Master Module

SIDM = Social Interactive Drawing Module

**Use Case 1: Load Master Module**

Actors: Robot, Operator

Entry Conditions: The Robot is available in the correct environment.

Exit Conditions: The master module has been loaded.

Flow of Events:

1)    Initiate **Robot Startup**

2)    Start to **Load Activities Master Module**

3)    Prepare to **Detect Subject**

Exceptions:

            2a) The AMM cannot be loaded

            3a) The Robot cannot detect a subject within the allotted time

**Use Case 1.1: Robot Startup**

Actors: Robot, Operator

Entry Conditions: The Robot is available in the correct environment.

Exit Conditions: The Robot has completed its startup sequence.

Flow of Events:

1)    The NAO button is pressed by the Operator

2)    The Robot turns on

3)    The Robot performs its standard startup sequence

Exceptions:

2a) The Robot does not turn on

3a) The Robot performs its software update sequence

* Action: Wait for the software update to complete and normal startup to resume.

3b) The Robot does not perform its standard startup sequence

**Use Case 1.2: Load Activities Master Module**

Actors: Robot, Operator

Entry Conditions: The Robot has been turned on and completed its startup sequence.

Exit Conditions: The AMM has been loaded.

Flow of Events:

1. The NAO system loads the AMM
2. The AMM self-initialization sequence is started
3. The Robot gives a verbal notification that the module has been loaded and initialized (**Speak Interaction Statement**)
4. The Robot pauses to allow the Operator to position it and leave the environment (assuming Operator is not Subject)
5. The AMM awaits the detection of a subject

Exceptions:

            1a) The NAO system fails to load the AMM

            1b) The NAO system cannot locate the AMM

            2a) The AMM fails during its self-initialization sequence

**Use Case 1.3: Detect Subject**

Actors: Robot, Subject

Entry Conditions: The activities master module has been loaded and its initial sequence begun.

Exit Conditions: The Subject has been detected.

Flow of Events:

1)    The Subject enters the environment

2)    The Robot detects the Subject using its camera and audio systems

3)    The Robot turns its head to look at the Subject

4)    An event is raised to notify the AMM that the Subject has been detected

Exceptions:

            1a) The Subject does not enter the environment in the allotted amount of time

            2a) The Robot fails to detect a Subject within the environment

            3a) The Robot does not turn its head to look at the Subject

            3b) The Robot turns its head, but does not look at the Subject directly

**Use Case 2: Initialize Activities Master Module**

Actors: Robot, Subject

Entry Conditions: The AMM has been loaded and the Subject has been detected.

Exit Conditions: The Subject has been greeted and all activity modules have been discovered.

Flow of Events:

1)    The AMM receives the Subject detected event

2)    The Robot greets the Subject (**Greet Subject**)

3)    The AMM **Discovers Available Activity Modules**

Exceptions:

            1a) The AMM does not receive the Subject detected event in the allotted time

            3a) The Robot cannot locate any available Activity Modules

**Use Case 2.1: Greet Subject**

Actors: Robot, Subject

Entry Conditions: The Subject has been detected.

Exit Conditions: The Subject has been greeted.

Flow of Events:

1)    Select a greeting (**Select Interaction Statement**)

2)    Speak greeting to the Subject (**Speak Interaction Statement**)

3)    Invite the Subject to approach (**Speak Interaction Statement**)

Exceptions:

N/A

**Use Case 2.2: Discover Available Activity Modules**

Actors: Robot

Entry Conditions: The activities master module has been loaded.

Exit Conditions: All activity modules have been discovered.

Flow of Events:

1)    The AMM scans for Activity Modules

2)    The AMM loads the descriptions for each module found

Exceptions:

            1a) The AMM cannot locate any Activity Modules

            2a) An Activity Module is missing a description

**Use Case 3: Start Activity**

Actors: Robot, Subject

Entry Conditions: The activities master module has been initialized.

Exit Conditions: An activity module has been selected, loaded, and its initial operation sequence begun.

Flow of Events:

1. The Robot **Explains Commands**
2. The Robot **Explains Activities**
3. The Robot allows the Subject to **Select Activity**
4. The Robot **Loads the Selected Activity Module**
5. The activity module initialization sequence is started (**Start Activity Module Initialization Sequence**)

Exceptions:

            4a) An activity selection is not heard within the allotted time

            5a) An invalid selection is heard

* Action: **Handle Invalid Activity Selection**

**Use Case 3.1: Explain Commands**

Actors: Robot, Subject

Entry Conditions: The activities master module has been initialized.

Exit Conditions: The available master-level commands have been explained to the subject.

Flow of Events:

1. Speak instructions for each master-level command (**Speak Interaction Statement**)

(Note: List of master-level commands to be defined)

Exceptions:

N/A

**Use Case 3.2: Explain Activities**

Actors: Robot, Subject

Entry Conditions: The activities master module has been initialized and all activity modules discovered.

Exit Conditions: The available activities modules have been explained.

Flow of Events:

1. Speak the name of each activity module and its description (**Speak Interaction Statement**)

Exceptions:

N/A

**Use Case 3.3: Select Activity**

Actors: Robot, Subject

Entry Conditions: The available activities have been explained to the subject.

Exit Conditions: The subject has selected a valid activity.

Flow of Events:

1. Prompt the subject to speak the name of an activity (**Speak Interaction Statement**)
2. **Listen for Selection** of an activity
3. The Robot **Validates the Activity Selection**

Exceptions:

3a) The Subject selects an invalid activity

* Action: **Handle Invalid Activity Selection**

3b) The Subject does not select an activity within the allotted time

**Use Case 3.4: Listen for Selection**

Actors: Robot, Subject

Entry Conditions: The activities master module has been initialized, the Subject identified, and the activity modules list spoken to the Subject.

Exit Conditions: The Robot has heard a potential selection or command.

Flow Events:

1. The Robot waits to hear a response from the Subject (**Listen for Question Response**)
2. The Robot forwards a complete statement to the appropriate modules for processing

Exceptions:

N/A

**Use Case 3.5: Validate Activity Selection**

Actors: Robot

Entry Conditions: The Robot has heard a potential activity selection.

Exit Conditions: The Robot has identified a potential activity selection as valid or invalid.

Flow of Events:

1. The Robot evaluates the Subject’s statement
2. The Robot identifies a valid activity selection
3. The Robot forwards the activity selection to the AMM to be loaded

Exceptions:

2a) The Robot does not identify a valid activity selection

Action: **Handle Invalid Activity Selection**

**Use Case 3.6: Handle Invalid Activity Selection**

Actors: Robot, Subject

Entry Conditions: The robot has identified an invalid activity selection.

Exit Conditions: The robot has notified the user of the invalid selection and prompted for the next action to be taken.

Flow of Events:

1. The Robot verbally notifies the Subject that it could not identify a valid activity selection (**Speak Interaction Statement**)
2. The Robot asks the Subject whether they wish to hear the activity module list again (Speak Interaction Statement)
3. If the Subject wishes to hear the activity module list, the Robot **Explains Activities** again
4. The Robot prompts the Subject to **Select Activity**

Exceptions:

N/A

**Use Case 3.7: Load Selected Activity Module**

Actors: Robot

Entry Conditions: The robot has identified a valid activity selection.

Exit Conditions: The activity module has been loaded.

Flow Events:

1. The activity module is loaded
2. All necessary module resources are initialized/allocated

Exceptions:

N/A

**Use Case 3.8: Start Activity Module Initialization Sequence**

Actors: Robot

Entry Conditions: An activity module has been loaded.

Exit Conditions: The activity module’s initialization sequence has begun.

Flow Events:

1. The module initialization sequence is started

Exceptions:

N/A

**Use Case 4: Select Object to Draw**

Actors: Robot, Subject

Entry Conditions: The Social Interactive Drawing Module has been loaded and initialized.

Exit Conditions: A valid object has been selected by the subject for the robot to draw.

Flow Events:

1. Verbally explain drawing activity (**Speak Interaction Statement**)
2. The Robot **Lists Drawable Objects**
3. The Robot **Prompts the Subject to Select Object**
4. The Robot **Validates the Object Selection**

Exceptions:

4a) A selection is not heard by the robot

4b) An invalid selection is heard

* Action: **Handle Invalid Object Selection**

**Use Case 4.1: List Drawable Objects**

Actors: Robot, Subject

Entry Conditions: The Social Interactive Drawing Module has been loaded and initialized.

Exit Conditions: The available objects that can be drawn have been listed to the subject.

Flow Events:

1. Scan for available drawable object instructions
2. List all available drawable objects to Subject (**Speech Interaction Statement**)

Exceptions:

**Use Case 4.2: Prompt Object Selection**

Actors: Robot, Subject

Entry Conditions: The robot has listed all available objects which can be drawn to the subject.

Exit Conditions: The robot has prompted the subject to select an object to draw.

Flow Events:

1. Prompt the Subject to select one of the objects listed (**Speech Interaction Statement**)

Exceptions:

**Use Case 4.3: Validate Object Selection**

Actors: Robot

Entry Conditions: The robot has heard a potential object selection.

Exit Conditions: The robot has identified whether the object selection is valid or invalid.

Flow Events:

1. Listen to the Subject state one of the objects listed
2. Make sure object that the Subject selected is one of available objects

Exceptions:

1a) A selection is not heard by the robot

2a) A selection is not one of valid selections

* Action: **Handle Invalid Object Selection**

**Use Case 4.4: Handle Invalid Object Selection**

Actors: Robot, Subject

Entry Conditions: The robot has identified an invalid object selection.

Exit Conditions: The robot has notified the subject of the invalid selection and prompted for the next action to be taken.

Flow Events:

1. The Robot tells the Subject that their selection is invalid (**Speech Interaction Statement**)
2. The Robot **Prompts the Subject to Select Object**

Exceptions:

**Use Case 5: Obtain Writing Implement**

Actors: Robot, Subject

Entry Conditions: The robot has been instructed to draw a valid object.

Exit Conditions: The robot has obtained a writing implement which is oriented correctly for drawing.

Flow Events:

1. Robot asks Subject which writing implement to draw a picture (**Interact with Subject)**
2. Robot finds and picks up a writing implement
3. Robot verifies and confirms Subject’s selection of writing implement
4. Robot draws a picture (**Draw Object**)

Exceptions:

2a) Robot cannot find a writing implement

2b) Robot cannot hold a writing implement (**Evaluate Marker Orientation**)

**Use Case 5.1: Determine if Holding Marker**

Actors: Robot

Entry Conditions: The robot has been instructed to draw a valid object.

Exit Conditions: The robot has determined whether it is already holding a marker or not.

Flow Events:

1. Robot scans the environment and finds a marker
2. Robot picks up the marker
3. Robot determines if it is holding a marker correctly (**Evaluate Marker Orientation)**

Exceptions:

1a) Robot scans environment and does not see a marker

**Use Case 5.2: Ask Whether to Use Current Marker**

Actors: Robot, Subject

Entry Conditions: The robot has determined that it is already holding a marker.

Exit Conditions: The robot has determined whether it should continue using the current marker or obtain a new one.

Flow Events:

1. Robot holds current marker from Subject’s selection
2. Robot verifies and confirms current marker with Subject (**Validate Marker Selection**)
3. Robot draws a picture with the marker selected from the Subject

Exceptions:

2a) Subject changes marker selection (**Validate Marker Selection**)

**Use Case 5.3: Validate Marker Selection**

Actors: Robot

Entry Conditions: The robot has heard a potential marker selection instruction.

Exit Conditions: The robot has identified whether the selection instruction statement is valid or invalid.

Flow Events:

1. Robot listens to Subject’s selection of marker to color the picture
2. Robot asks Subject which marker to color the picture (**Interact with Subjec**t)
3. Robot determines if Subject’s selection of marker is valid
4. Robot finds and picks the marker selected from the Subject

Exceptions:

3a) Subject’s choice of marker is invalid for the Robot

**Use Case 5.4: Drop Marker**

Actors: Robot

Entry Conditions: The robot needs to obtain a new marker but is currently holding one.

Exit Conditions: The robot is no longer holding the marker.

Flow Events:

1. Robot holds and draws with the current marker
2. Robot finishes drawing a picture with the current marker (**Complete Activity**)
3. Robot drops the marker on the floor

Exceptions:

1a) Robot cannot hold a new marker correctly (**Evaluate Marker Orientation)**

2a) Robot does not finish drawing with current marker (**Request New Marker**)

**Use Case 5.5: Request New Marker**

Actors: Robot, Subject

Entry Conditions: The robot has determined that it needs to obtain a new marker and is not currently holding a marker.

Exit Conditions: The robot has requested the subject to provide a new marker and the subject has placed the marker in its hand.

Flow Events:

1. Robot draws a picture with a marker selected from the Subject
2. Robot asks Subject to continue drawing with current marker (**Interact with Subject**)
3. Robot determines whether Subject wants to change current marker to a new marker

Exceptions:

3a) Subject does not want to change his/her selection of the current marker

**Use Case 5.6: Evaluate Marker Orientation**

Actors: Robot

Entry Conditions: The robot has a marker in its hand.

Exit Conditions: The robot has determined whether the marker is oriented correctly for drawing.

Flow Events:

1. Robot picks up a marker with one hand with a full-fist orientation
2. Robot determines if marker’s orientation can be held by its hand

Exceptions:

2a) Marker’s orientation cannot be hold by Robot’s hand

**Use Case 5.7: Evaluate Whether Cap On/Off**

Actors: Robot

Entry Conditions: The robot has a marker in its hand which has been determined to be oriented correctly.

Exit Conditions: The robot has determined whether the marker’s cap is on or off.

Flow Events:

1. Robot is able to hold a marker (**Evaluate Marker Orientation**)
2. Robot examines the marker
3. Robot determines if marker’s cap is on or off (**Handle Marker Cap On**)

Exceptions:

3a) Marker’s cap is already off, Robot does not need to examine the marker

**Use Case 5.8: Handle Invalid Marker Orientation**

Actors: Robot, Subject

Entry Conditions: The robot has determined that it is holding a marker in an incorrect orientation for drawing.

Exit Conditions: The robot has notified the subject that the marker is incorrectly oriented and prompted for the next action to be taken.

Flow Events:

1. Robot is not able to hold the marker correctly (**Evaluate Marker Orientation**)
2. Robot notifies Subject it cannot hold the marker correctly
3. Robot asks Subject to select a different marker to draw (**Interact with Subject**)

Exceptions:

2a) Robot can hold a marker correctly (**Evaluate Marker Orientation)**

**Use Case 5.9: Handle Marker Cap On**

Actors: Robot, Subject

Entry Conditions: The robot has determined that it is holding a marker with its cap on.

Exit Conditions: The robot has prompted the subject to remove the cap and the subject has done so.

Flow Events:

1. Robot can hold the marker upward
2. Robot examines and determine marker’s cap is on or off (**Evaluate Whether Cap On/Off**)
3. Robot asks Subject to remove the marker’s cap if it is on the marker

Exceptions:

2a) Marker’s cap is already removed (**Evaluate Whether Cap On/Off)**

3a) How tight the marker is on for the Subject to remove

**Use Case 6: Draw Object**

Actors: Robot, Subject

Entry Conditions: The robot has been instructed to draw a valid object and is holding a writing implement properly for drawing.

Exit Conditions: The robot has drawn the requested object on the drawing surface.

Flow Events:

1. The robot identifies the drawing surface and boundaries.
2. The robot draws the selected picture on the drawing surface.
3. The robot interacts with the user in the event of any errors.

Exceptions:

1a) The robot cannot locate or access drawing surface or identify its boundaries.

**Use Case 6.1: Locate Drawing Surface**

Actors: Robot, Subject

Entry Conditions: The robot has been instructed to draw a valid object and is holding a writing implement properly for drawing.

Exit Conditions: The robot has located a drawing surface.

Flow Events:

1. The robot scans the environment for the drawing surface.

Exceptions:

1a) The robot does not find the drawing surface.

**Use Case 6.2: Determine Drawing Surface Boundaries**

Actors: Robot

Entry Conditions: The robot has located a drawing surface.

Exit Conditions: The robot has identified the boundaries of the drawing surface.

Flow Events:

1. The robot has located the drawing surface.
2. The robot scans the drawing surface to determine its boundaries.

Exceptions:

2a) The robot is unable to locate the boundaries.

**Use Case 6.3: Evaluate Drawing Surface Accessibility**

Actors: Robot

Entry Conditions: The robot has located a drawing surface and identified its boundaries.

Exit Conditions: The robot has evaluated whether a drawing surface is accessible or inaccessible.

Flow Events:

1. The robot has determined and stored the boundary locations of the drawing surface.
2. The robot determines if the surface is accessible.

Exceptions:

2a) The robot cannot access the drawing surface.

**Use Case 6.4: Activate Advanced Motor Control Module**

Actors: Robot

Entry Conditions: The robot has identified an accessible drawing surface.

Exit Conditions: The robot has activated the advanced motor control module.

Flow Events:

1. The robot has determined the drawing surface to be accessible.
2. The robot loads the module for advanced motor control.

Exceptions:

N/A

**Use Case 6.5: Execute Object Drawing Instructions**

Actors: Robot

Entry Conditions: The robot has been instructed to draw a valid object and has activated the advanced motor control module.

Exit Conditions: The robot has drawn the instructed object on the drawing surface.

Flow Events:

1. The robot has successfully loaded the advanced motor control module.
2. The robot executes the correct drawing instructions for the shape specified by the user previously.

Exceptions:

N/A

**Use Case 6.6: Handle Inaccessible Drawing Surface**

Actors: Robot, Subject

Entry Conditions: The robot has located a drawing surface but has determined that is inaccessible.

Exit Conditions: The robot has notified the subject of the inaccessible drawing surface and prompted the subject to position it properly.

Flow Events:

1. The robot determines the drawing surface to be inaccessible.
2. The robot notifies the user of an inaccessible drawing surface and prompts the user to make adjustments.
3. The robot waits for the user to make the surface accessible and give a command to resume or for the user to terminate the program.

Exceptions:

N/A

**Use Case 6.7: Handle No Drawing Surface**

Actors: Robot, Subject

Entry Conditions: The robot has been unable to locate a drawing surface in the environment.

Exit Conditions: The robot has notified the subject that it cannot locate a valid drawing surface and has prompted the subject to select the next action.

Flow Events:

1. The robot determines that it cannot find the drawing surface.
2. The robot notifies the user that it cannot find the drawing surface and prompts the user to make adjustments.
3. The robot waits for the user to make adjustments and give a command.

Exceptions:

  N/A

**Use Case 7: Interact with Subject**

Actors: Robot, Subject

Entry Conditions: The robot has determined that it needs to interact with the subject.

Exit Conditions: The robot has completed the interaction with the subject.

Flow Events:

1. The Robot **Selects an Interaction Statement**
2. The Robot **Speaks the Interaction Statement**
3. If the statement was a question, the Robot **Listens for Question Response**

Exceptions:

N/A

**Use Case 7.1: Select Interaction Statement**

Actors: Robot

Entry Conditions: The robot has determined that it needs to speak to the subject.

Exit Conditions: The robot has selected an interaction statement to speak to the subject.

Flow Events:

1. Communication providers within the NAO system are polled to determine the appropriate handler
2. The appropriate handler assesses the environment and system state
3. The handler generates an interaction statement

Exceptions:

1a) No communication handler is available to handle the current situation

**Use Case 7.2: Speak Interaction Statement**

Actors: Robot, Subject

Entry Conditions: The robot has selected an interaction statement to speak to the subject.

Exit Conditions: The robot has spoken the interaction statement to the subject.

Flow Events:

1. The NAOqi Text-to-Speech module receives a posted instruction containing text to speak
2. The module processes the text and speaks the resulting message

Exceptions:

N/A

**Use Case 7.3: Listen for Question Response**

Actors: Robot, Subject

Entry Conditions: The robot has asked the subject a question.

Exit Conditions: The robot has heard a possible response to the question.

Flow Events:

1. The Robot’s microphone’s collect audio data
2. Collected audio data is processed to construct a linguistic statement
3. Identified linguistic statements are posted to the AMM communications broker
4. The AMM communications broker **Evaluates the Question Response**
5. A communication handler triggers the appropriate actions to handle the question response

Exceptions:

N/A

**Use Case 7.4: Evaluate Question Response**

Actors: Robot

Entry Conditions: The robot has heard a possible response to a question.

Exit Conditions: The robot has evaluated a subject’s response to a question and determined the next action to take.

Flow Events:

1. The AMM communications broker assesses whether the statement is a master level command
2. If the AMM cannot identify a master level command, the statement is forwarded to each communication handler in order of most recent registration until the statement is handled

Exceptions:

  2a) No communication handler is able to handle the statement

**Use Case 8: Complete Activity**

Actors: Robot, Subject

Entry Conditions: The robot has completed the sequence for an activity.

Exit Conditions: The robot has determined whether to exit or restart the activity.

Flow Events:

1. The Robot assesses whether an activity is repeatable
2. If the activity is not repeatable, the AMM **Exits the Activity Module**
3. Otherwise, the Robot **Prompts Whether to Continue the Activity**
4. If the Subject elects to continue, the AMM **Restarts the Activity Module**

Exceptions:

N/A

**Use Case 8.1: Prompt Whether to Continue Activity**

Actors: Robot, Subject

Entry Conditions: The Robot has recognized some condition indicating that an activity may need to be ended or restarted.

Exit Conditions: The robot has prompted the subject to select the next action.

Flow Events:

1. The Robot notifies the Subject that an activity may need to be ended (**Speak Interaction Statement**)
2. The Robot asks whether the Subject is ready to end the current activity (**Speak Interaction Statement**)
3. The Robot listens for the Subject’s response (**Listen for Question Response**)
4. If the Subject has selected to end the activity, the AMM **Exits the Activity Module**, otherwise the activity is resumed
5. If the activity has been exited, the Robot prompts the Subject to **Select an Activity**

Exceptions:

2a) The current activity must be ended

* Action: **Exit Activity Module**

**Use Case 8.2: Restart Activity Module**

Actors: Robot

Entry Conditions: The robot has determined that an activity module needs to be restarted.

Exit Conditions: The robot has started the activity module’s initialization sequence.

Flow Events:

1. The AMM clears the current activity state
2. The AMM runs the activity module’s initialization sequence

Exceptions:

N/A

**Use Case 8.3: Exit Activity Module**

Actors: Robot

Entry Conditions: The robot has determined that it needs to exit the current activity module.

Exit Conditions: The robot has exited the current activity module and returned to the activity selection stage.

Flow Events:

1. The AMM clears the current activity state
2. The AMM unloads the activity module

Exceptions:

N/A