WRS Partner Robot Challenge (Virtual Space) 2018

WRS Partner Robot Challenge (Virtual Space) is the world's first competition held in a cyber-physical environment.

1. Introduction

1.1. Competition concept

The Partner Robot Challenge (Virtual Space) evaluates how well intelligent robots can engage in natural and friendly communication with users and achieve various support behaviors in daily environments. The competition is designed on the basis of the SIGVerse simulator, which enables robots to make embodied and social interactions in virtual reality (VR) environments.
This competition consists of four tasks, the rules of which are explained in detail in the following sections. This document mainly explains the rules and the evaluation method. The software configuration and communication protocols used in the software system are detailed in GitHub repository of this competition: https://github.com/PartnerRobotChallengeVirtual

1.2. Organization

1.2.1. Executive committee

The executive committee is responsible for organizing the Partner Robot Challenge (Virtual Space)

Hiroyuki Okada (Tamagawa University, Japan)

Tetsunari Inamura (National Institute of Informatics, Japan)

1.2.2. Technical committee

The technical committee is responsible for providing software modules for the competition.

Tetsunari Inamura (National Institute of Informatics, Japan)

Yoshiaki Mizuchi (National Institute of Informatics, Japan)

Hiroki Yamada (National Institute of Informatics, Japan)

1.3. Fair Play

Fair play and cooperative behavior are expected from all teams during the entire competition. This includes:

• not trying to cheat (e.g. pretending autonomous behavior exists where there is none),

• not trying to exploit the rules (e.g. not trying to solve the task but trying to score), and

• not trying to interrupt test subjects in the Human Navigation task.

Disregard of these rules will lead to penalties in the form of negative scores and disqualification for a test or even for the entire competition.

1.4. Troubleshooting

All the competition tasks require repetition of several sessions. If the n-th session terminated due to a team's bug, the n-th session score will be zero and the team will be able to restart only from the subsequent n+1-th session. If the trouble is not a team's bug, the team can restart from the current n-th session. The cause of the trouble will be determined by the technical committee. The team should cooperate with the technical committee to investigate any trouble, such as by submitting ROS log files and source code.

Basically, if trouble arising from software bugs is found, the technical committee will investigate the problem status and the executive committee will judge how to deal with the trouble.

1.5. Communication for competition operation

The executive and technical committees will often make announcements to organize and operate the competition. The official communication tool between the teams and executive/technical committees will be Slack, so the teams should always check Slack for announcements. Communication by e-mail is not encouraged.

2. Handyman

2.1. Background & Concept

This task evaluates the fundamental abilities of the robot to support people's daily activities. The main evaluation target will be navigation with obstacle avoidance, object recognition, object manipulation, and human detection. Additionally, the robot will sometimes need to detect a human's misunderstanding and clarify the error in the instruction. For example, a person may mistakenly instruct the robot to find a non-existing object. To detect the error, the robot has to check the environment and determine that the reason it cannot find the object is not that the robot software overlooks it but that the human's instruction contains an error. This function is also very important for a partner robot.

2.2. System Configuration

The system configuration for the Handyman task is depicted in Fig.2.1.

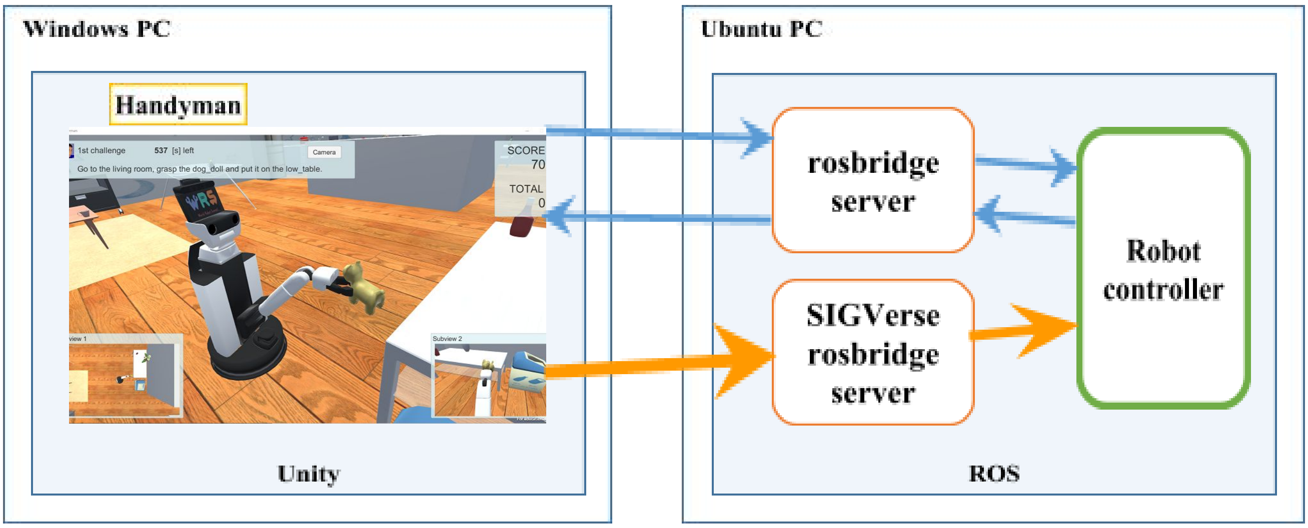

Figure 2.1 System configuration for Handyman

The Windows PC executes the Handyman software on the basis of Unity and SIGVerse. The Ubuntu PC executes the rosbridge server, the SIGVerse rosbridge server, and the robot controller implemented by the team.

The Handyman software and the robot controller communicate through the basic rosbridge server, but communication containing a large amount of data (e.g., sensor data) is transmitted through the SIGVerse rosbridge server.

The virtual robot should be controlled by the robot controller on the Ubuntu PC when the instructions from the human avatar are sent to the robot. The robot controller should send ROS messages such as Twist and JointTrajectory to the Handyman software to control the robot.

The Handyman software sends JointState, TF, sensor information, and other ROS messages at regular intervals to the robot controller.

2.3. Task Flow

The detailed flow of the handyman task is described at GitHub wiki at the following URL:
https://github.com/PartnerRobotChallengeVirtual/handyman-unity/wiki/SystemOverview

2.4. Time Limits

- The time limit for each session is *N* minutes.
- The competition consists of *M* sessions.
- *N* and *M* are announced at least 7 days before the competition. The range are $6 \leqq N \leqq 10$ and $10 \leqq M \leqq 24$.
- The timer starts when the sensor signals are distributed to the robot.

2.5. Score (in each session)

Action	Score
The robot arrives at the instructed room	*20*
The robot fails to arrive at the instructed room	*-10*
The robot grasps the instructed object	*50*
The robot fails to state that the grasping is completed	*-10*
The robot achieves the instructed carrying behavior	*30*
The robot fails to state that the carrying behavior is completed	*-10*
The robot finds human's error and succeeds at suggesting the correction	*50*
The robot fails to suggest the correction to the human's error	*-10*
The robot collides with objects (each time)	$-\alpha$
The object grasped/released by the robot collide with other objects (each time)	$-\beta$
Total score (excluding penalties)	***150***

The score will be 0 points even if many deductions result in an actual score below zero.

2.5.1. Penalties for collisions

If the robot collides with an object in the environment, a $-\alpha$ point penalty will be given.

(1) If the collided with object is a graspable object on the floor (i.e., the base of the robot collides with a graspable object):

α will be 5 (i.e., a -5 point penalty will be given).

Duplicated penalties for the same object are never given. (The penalty will be given only once for each unique object.)

(2) If the collided with object is furniture or a graspable object not on the floor:

α will be $5 \leqq \alpha \leqq 50$. The detailed value will be calculated by the collision impact as explained below. Duplicated penalties will be given for each collision.

If an object grasped or released by the robot collides with another object, a $-\beta$ point penalty will be

given. For example, a released object dropping onto another object will cause this penalty. Sometimes, collision might be detected just after the robot releases an object. In that case, the collision will be ignored.
Except for case (1), α and β are calculated with a consideration of the impact of the collision each time. α and β are proportional to the collision velocity. Their ranges are $5 \leqq \alpha \leqq 50$ and $1 \leqq \beta \leqq 50$. In order words, the max value of the collision penalty is 50 points each time.

2.6. Remarks

2.6.1. Instruction to robot

Instruction statements are given as natural language expressions in English, described in text (sequence of ASCII characters). The instruction always includes movement, grasping, and carrying behavior such as:

Go to the xxx, grasp the YYY and ZZZ,

where xxx indicates the room, YYY indicates the target object, and ZZZ indicates the carrying behavior. The carrying behavior consists of 1) put the target object in/on a certain destination position, 2) discard the object into a trash can, and 3) hand the object over to the avatar.
The names of the objects and furniture are fixed and will be announced at least 14 days before the competition. To express the target position, the following candidate phrases may be used: 'next to the x', 'on the x', 'in the x', 'under the x', 'close to the x', 'between x and y' and 'in front of x'. The 'x' and 'y' are landmarks for the target object that should be the names of objects or furniture. A landmark for the target object can be a graspable or non-graspable object. These candidate phrases (prepositions and spatial relationships) might be increased or modified in an English correction process by native English speakers. If they are, the candidate phrases will be announced at least 14 days before the competition.

2.6.2. Carrying behavior (Handover to avatar)

The instruction statement might necessitate handing over a target object to the avatar. To achieve the handover action, the robot should extend its arm, which grasps the target object, to the avatar's chest. The end effector should be inside a sphere with a 30-cm radius where the central position is the chest of the avatar.

If the instructed carrying behavior is not to hand over but just to take a target object to a certain destination, the robot should put the target object in/on the destination gently so as not to generate a big impact.

2.6.3. Dynamic changes in environment

Another avatar who is not related to the instruction to the robot may stand or walk around in the environment. The avatar who instructs the robot also may walk around after the instruction. The robot should distinguish the unrelated avatar.

The positions of furniture such as desks, tables, and chairs are fixed; however, the trash cans might be located at positions different from those in the original room layout file, which will be made public beforehand.

The target object instructed by the avatar sometimes might not exist. In such a case, the robot should recognize and declare the object's non-existence. If the declaration is succeeded, additional points will be given, and the avatar should correct the instruction.

2.7. Screenshot

An example of the screen is shown in Fig.2.2.

Figure 2.2 A main window of Handyman (sample)

The meaning of each component is described at the GitHub wiki at the following URL:
https://github.com/PartnerRobotChallengeVirtual/handyman-unity/wiki/Screen

2.8. Room Layout and Furniture

Rooms used in the competition are listed in Table 2.2.
The actual room layout will be announced 24 hours before the competition in the Unity project files. At the same time, overhead views of the layout with the room area definition will be provided. Figure 2.3 shows an example of the room layout with the room area definitions. More than two layouts will be used. Each layout has a name such as ‘Layout_A’, which is included in the announcement.

Table 2.2 List of the name of the room

No	Name
1	bed room
2	kitchen
3	living room
4	lobby

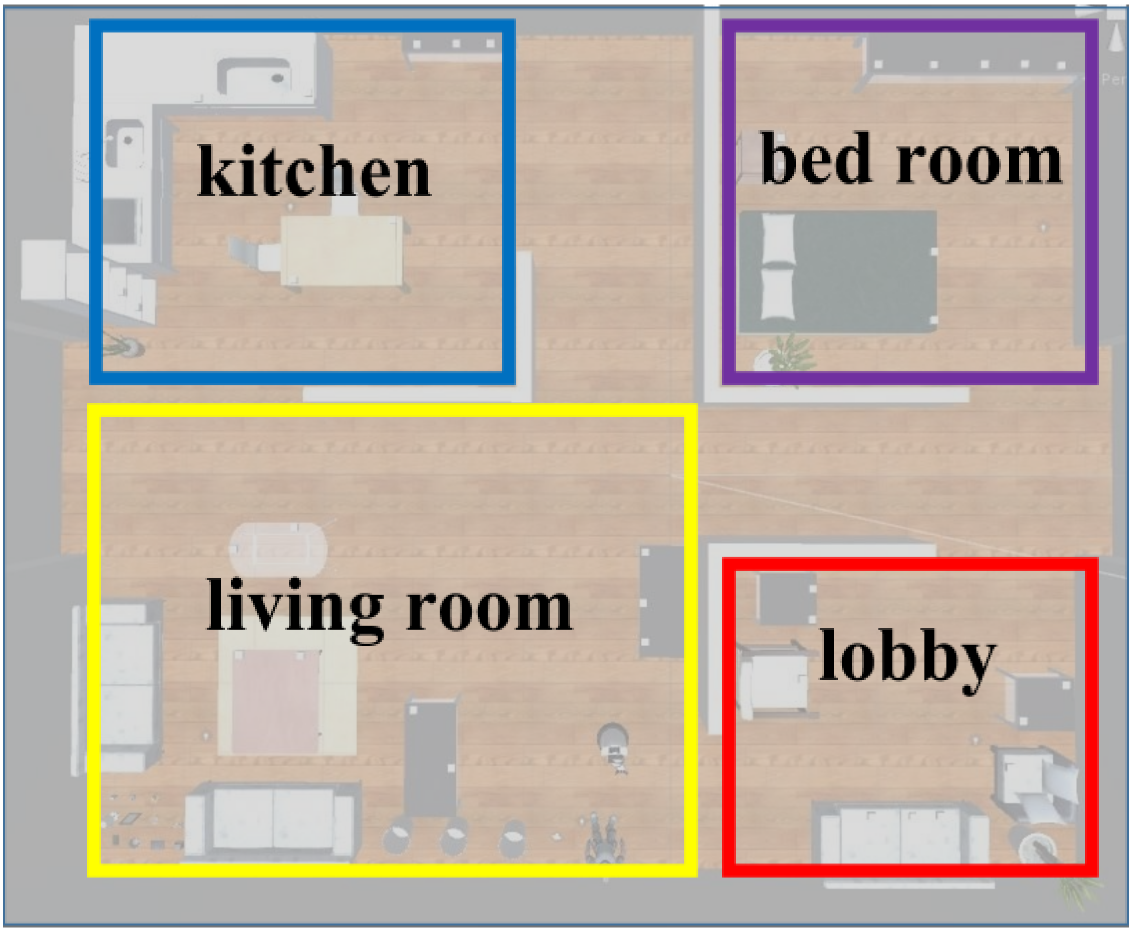

Figure 2.3 Example room layout

Each room area is separated by several walls. The whole of the robot base should be inside the area surrounded by the planes, which correspond to the inner side of the separating walls. After entering the target area, the robot should send a message to the avatar. The plane to define the boundary of the room area might be shifted ±3 cm from the actual wall location. Therefore, the robot is recommended to proceed to the central area of the room before sending the message.

The names of furniture are fixed and will be announced at least 14 days before the competition. An identical label is never used for different pieces of furniture. For example, just ‘table’ never used if there are a ‘kitchen table’, ‘wooden table’, ‘side table’, and so on.

2.9. Graspable Objects

The objects in the competition consist of graspable objects and non-graspable objects such as furniture. The robot can only grasp the graspable objects. Example graspable objects are shown in Table 2.3 and Fig. 2.4. The set of the object name and 3D model in the Unity environment will be announced at least 14 days before the competition. All the label names shall be made public and announced; however, the 3D models for several objects in the Unity environment will be hidden. We call such objects ‘a-like objects’. The robot should estimate the appearance of the object using the label information.

The robot can grasp any object if the end effector is placed at an appropriate position and in an appropriate direction.

Table 2.3 Graspable object list (example).

Note that these labels are just examples. The final labels might be different.

No	Label	No	Label
1	apple	7	sugar
2	toy_penguin	8	soysauce
3	rabbit_doll	9	sauce
4	bear_doll	10	ketchup
5	dog_doll	11	tumbler
6	canned_juice	12	cup

Figure 2.4 Graspable objects (example)

2.10. Destinations

The destination is either on furniture or in a trash can when the carrying behavior is not to hand over to the avatar. Examples of the destination objects used in the task are shown in Table 2.4 and Fig. 2.5. The destination object list will be announced at least 14 days before the competition.

Table 2.4 Destination list (example)

No	Label	No	Label
1	low_table	4	trash_box_for_recycle
2	wagon	5	trash_box_for_burnable
3	white_table	6	trash_box_for_bottle_can

Figure 2.5 Destination objects (example)

2.11. System Setup & Protocol of Competition

2.11.1. Schedule in the preparation phase

- at least 7 days before the competition: *N* and *M* in section 2.4 will be announced.
- at least 7 days before the competition: Detail time schedule of the competition and real-world time limit will be announced.
- at least 14 days before the competition: Object list in Unity project files will be opened.
- at least 24 hours before the competition: Layout will be made public in Unity project files.
- *n* hours before the competition: Teams will set up the robot controller PC and stop modifying its software. The value n will be announced at least 7 days before the competition. The range of n will be $0 \leqq n \leqq 3$.

2.11.2. Computer settings

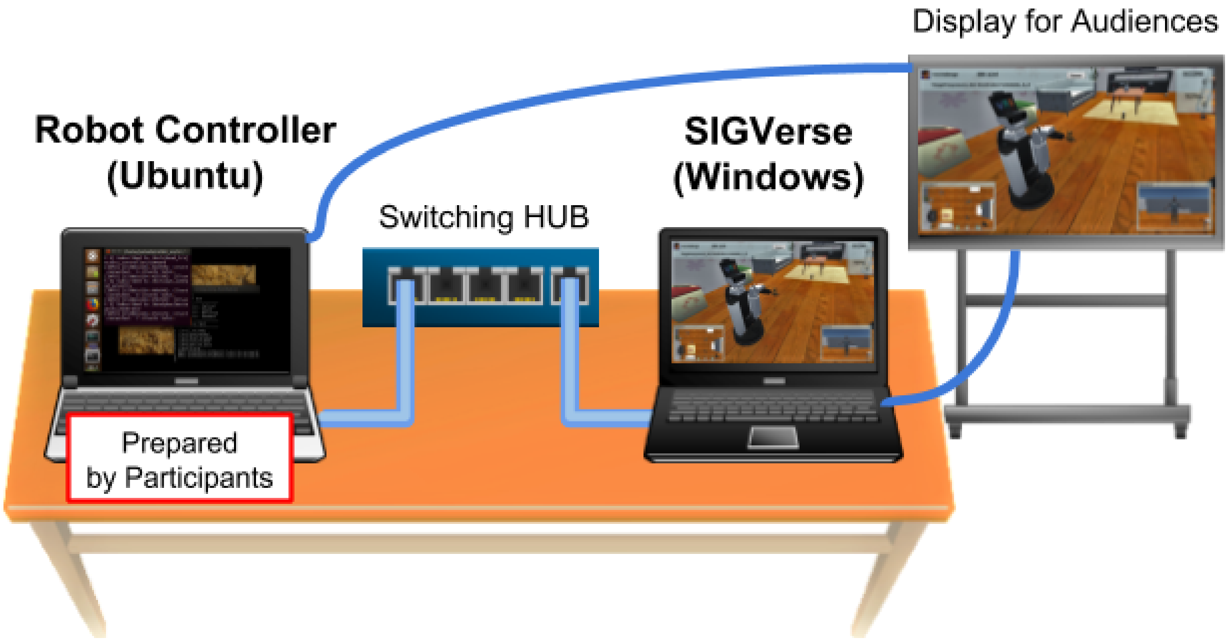

Figure 2.6 System overview for Handyman

The team should bring the robot controller PC to the table at least *n* hours before the competition. After setting it up, the team cannot modify any software modules. The team can display the PC screen on an official display to show the internal state of the robot to the audience.

2.11.3. Protocol of software execution

(1) The team executes the robot controller at the start time of the competition.

(2) The technical committee executes the Handyman software after the execution of the robot controller.

If software trouble occurs, the technical committee will deal with the trouble as described in Section 1.4. If this happens, the time counter in the virtual software will be paused; however, the team should finish all sessions within the real-world time limit. If the team's software cannot complete all the sessions within the real-world time limit, the total score will be the accumulated score up to and including the session in which the time limit expires. The real-world time limit will be announced at least 7 days before the competition. The real-world time limit is never shorter than N x M minutes.

3. Interactive Cleanup

3.1. Background & Concept

This task evaluates the ability of the robot to understand pointing gestures. The same as in the Handyman Task, the required basic functions include navigation with obstacle avoidance, object recognition, object manipulation, human detection, and so on. Additionally, the robot has to pay attention to not only the instructions but also the behavior of the human avatar. Since the pointing gesture motions are fairly distributed to all the teams and several sessions will be repeated, we can evaluate the robot's performance statistically.

3.2. System Configuration

The system configuration for this competitive challenge is depicted in Fig. 3.1.

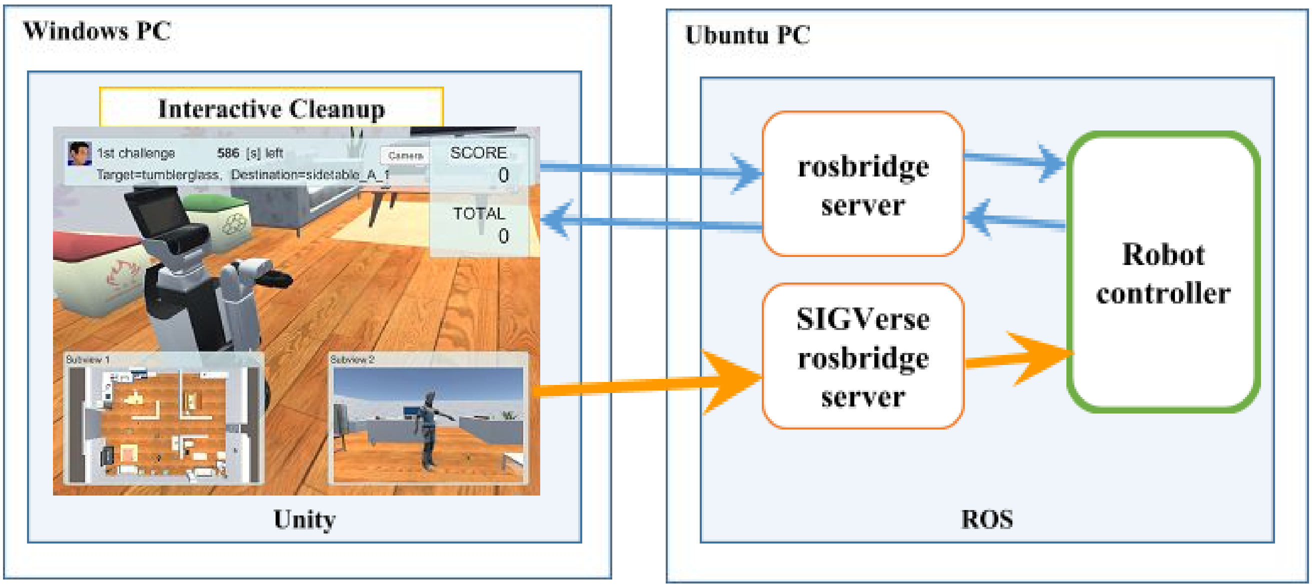

Figure 3.1 System configuration of Interactive Cleanup

The Windows PC runs the Interactive Cleanup program, which has been created in Unity. The Ubuntu PC runs the rosbridge server, the SIGVerse rosbridge server, and the robot controller created by the competition participants.

The Interactive Cleanup program and robot controller communicate through the basic rosbridge server, but communication containing a large amount of data (e.g., sensor data) is transmitted through the SIGVerse rosbridge server.

In the Interactive Cleanup program, robots move in accordance with the instructions from the robot controller when the human avatar issues a cleanup command to the robot. The instructions for the cleanup command are determined on the basis of the operations of and the messages sent by the human avatar.

The robot controller sends ROS messages such as Twist and JointTrajectory to the Interactive Cleanup program to move the robot in the Interactive Cleanup program.

The Interactive Cleanup program distributes JointState, TF, sensor information, and other ROS messages at regular intervals to the robot controller.

3.3. Task Flow

The flow of the Interactive Cleanup task is described at the GitHub wiki at the following URL:
https://github.com/PartnerRobotChallengeVirtual/interactive-cleanup-unity/wiki/SystemOverview

3.4. Time Limits

- The time limit for each session is *N* minutes.
- The competition consists of *M* sessions.
- *N* and *M* are announced at least 7 days before the competition. Their range are $6 \leqq N \leqq 10$ and $10 \leqq M \leqq 24$.
- The timer starts when the sensor signals are distributed to the robot.

3.5. Score (for each session)

Action	Score
The robot grasps the correct target object	*50*
The robot fails to grasp the target object	*-10*
The robot succeeds at disposing of or repositioning the target object	*50*
The robot fails to dispose of or reposition the target object	*-10*
The robot confirms whether an object is correct (each time)	*-10*
The robot requests a pointing gesture be repeated (each time)	*-10*
The robot collides with an object (each time)	$-\alpha$
The object grasped/released by the robot collides with other objects j(each time)	$-\beta$
Total score (excluding penalties)	***100***

The session score will be calculated as a summation of the above scores during the session (N minutes). If many penalties (negative scores) are given to the team and the accumulated score becomes negative, the session score will be 0.
The total score of the Interactive Cleanup will be calculated as a summation of all session scores.

3.5.1. Penalty for collisions

If the robot collides with an object in the environment, a $-\alpha$ point penalty will be given.
(1) If the collided with object is a graspable object on the floor (i.e., the base of the robot collides with a graspable object):
α will be 5 (i.e., a -5 point penalty will be given).
Duplicated penalties for the same object are never given. (The penalty will be given only once for each unique object.)
(2) If the collided with object is furniture or a graspable object not on the floor:
α will be $5 \leqq \alpha \leqq 50$. The detailed value will be calculated by the collision impact as explained below. Duplicated penalties will be given for each collision.

If an object grasped or released by the robot collides with another object, a $-\beta$ point penalty will be given. For example, a released object dropping onto another object will cause this penalty. Sometimes, collision might be detected just after the robot releases an object. In that case, the collision will be ignored.

Except for case (1), α and β are calculated with a consideration of the impact of the collision each time. α and β are proportional to the collision velocity. Their ranges are $5 \leqq \alpha \leqq 50$ and $1 \leqq \beta \leqq 50$. In order words, the max value of the collision penalty is 50 points each time.

3.6. Screenshot

An example of the screen is shown in Fig. 3.2.

Figure 3.2 Main window of Interactive Cleanup (sample)

The meaning of each component is described at the GitHub wiki at the following URL:
https://github.com/PartnerRobotChallengeVirtual/interactive-cleanup-unity/wiki/Screen

3.7. Room Layout

An example of a room used in the competition is shown below.

The layouts of the room used for the competition will be made public at least 24 hours before the competition in the Unity project files. The announcement procedure is the same as in the Handyman task.

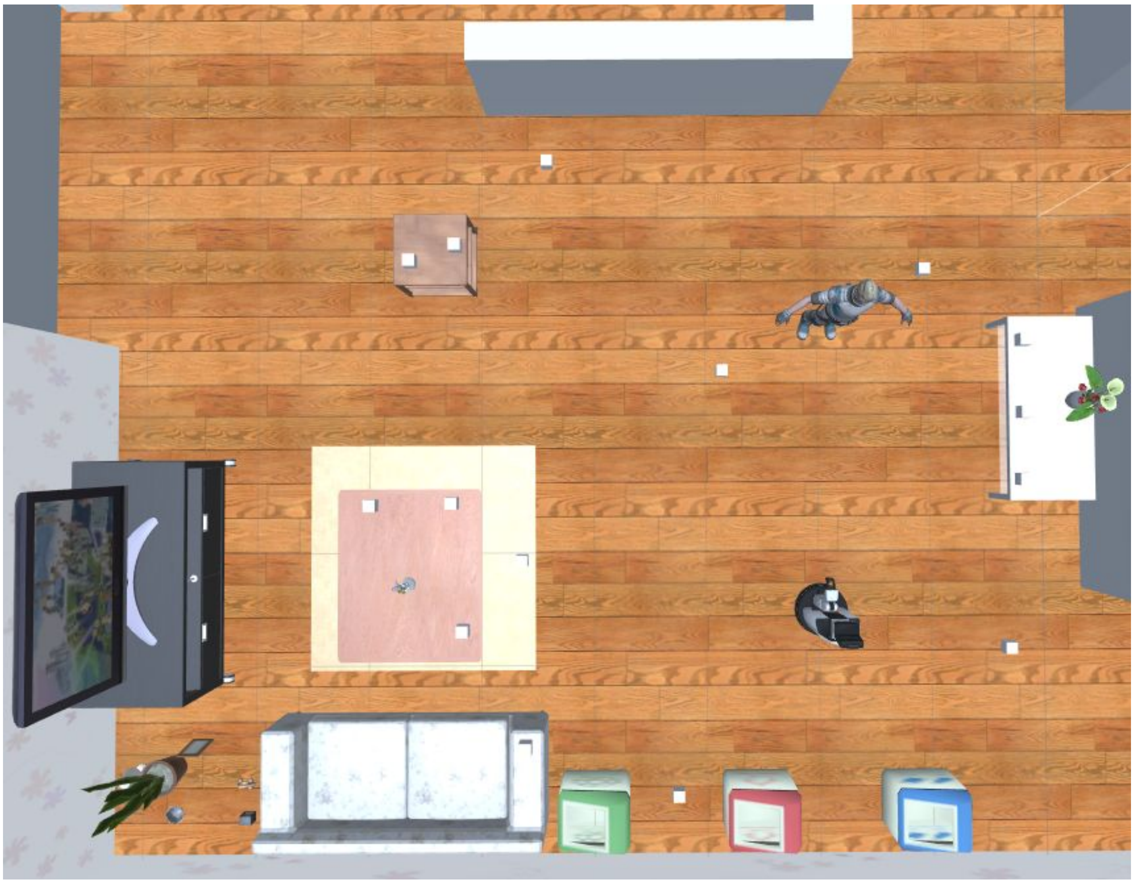

Figure 3.3 Example of room for Interactive Cleanup

3.8. Graspable Objects

Examples of target objects to be grasped by the robot are shown in Fig. 3.4. Final candidate objects will be announced at least14 days before the competition.

Figure 3.4 Graspable objects (example)

3.9. Destinations

The avatar instructs the robot where the target object should go by a second pointing gesture. The destination is either on furniture or in a trash can. Examples of the destination used in the task are shown in Fig.3.5. There are fewer than 15 candidate destinations. The destination object list will be announced at least 14 days before the competition.

Figure 3.5 Destination objects (example)

3.10. Remarks

3.10.1. Pointing gesture

The avatar performs two pointing gestures. The first and second pointing gestures identify the target object and its destination, respectively. The avatar maintains each posture/gesture for n seconds. For the pointing gesture, the avatar always uses the index finger. The side of the pointing gesture will be determined randomly for each session.

The range of the pause time duration is $2 \leqq n \leqq 10$.

3.10.2. Robot and avatar

The initial positions of the robot and the human avatar are always the same during all the sessions. The initial orientation of the robot will be set so that the robot can capture the human avatar image by the HeadCenterCamera. The robot should always be initially positioned and oriented where it can see the avatar in the initial camera frame.

The height of the human avatar never changes.

3.11. System Setup & Protocol of Competition

3.11.1. Schedule in preparation phase

● at least 7 days before the competition: *N* and *M* in section 3.4 will be announced.

● at least 7 days before the competition: Detailed time schedule of the competition and real-world time limit will be announced.

● at least 14 days before the competition: Object list will be made public in Unity project files.

at least 24 hours before the competition: Layout will be made public in Unity project files.

● n hours before the competition: Teams set up the robot controller PC and stop modifying its software

The value n will be announced at least 7 days before the competition. The range of n will be $0 \leqq n$

3.11.2. Computer settings

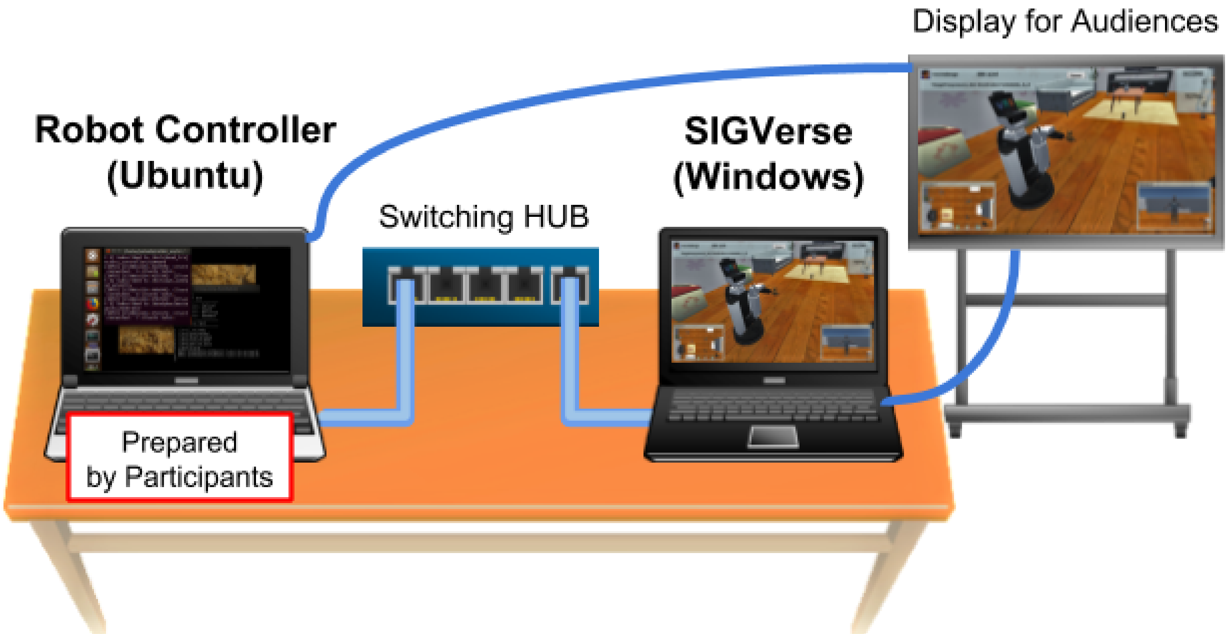

Figure 3.6 System overview for Interactive Cleanup

3.11.3. Protocol of software execution

(1) The team executes the robot controller at the start time of the competition.

(2) The technical committee executes the Interactive Cleanup program after the execution of the robot controller.

If software trouble occurs, the technical committee deals with the trouble as described in Section 1.4. If this happens, the time counter in the virtual software will be paused; however, the team should finish all the sessions within the real-world time limit. If the team's software cannot complete all the sessions within the time limit, the total score will be the accumulated score up to and including the session in which the time limit expires. The real-world time limit will be announced at least 7 days before the competition. The real-world time limit is never shorter than *N* x *M* minutes.

4. Human Navigation

4.1. Background & Concept

In the Human Navigation task, the robot has to generate friendly and simple natural language expressions to guide people to achieve several tasks in a daily environment. In order words, the roles of the robot and the user are the opposite of those in the Handyman task. The robot has to provide natural language instructions to tell a person to carry a certain target object to a certain destination such as, “Please take the cup on the table in front of you to the second drawer in the kitchen.” A person (test subject) logs in to an avatar in a VR space to follow the instructions. The test subject attempts to pick up the target object and take it to the destination using virtual reality devices in accordance with the instructions from the robot. The required time to complete the object transportation is measured and used to calculate points. The team that generates the easiest and most natural instructions for a person to follow will receive more points.

A reference video for the human navigation task is available from
https://drive.google.com/file/d/0BxjNl2PRT1F_WFpFMG1VZUFWY0U/view?usp=sharing

4.2. System Configuration

The computers and each program are connected in the configuration shown in Fig.4.1.

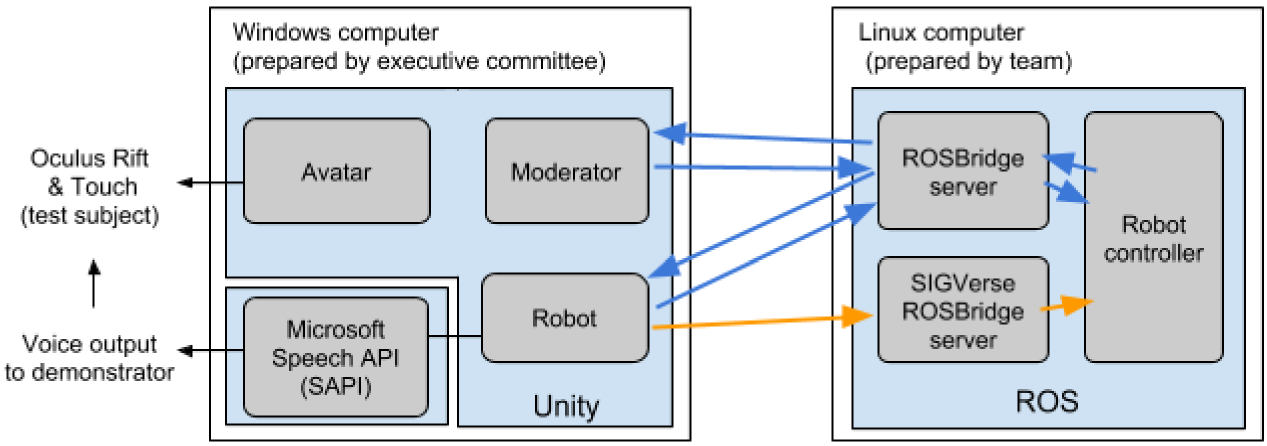

Figure 4.1: A System configuration of Human Navigation

4.3. Flow of task progress

The flow of the Human Navigation task is described at GitHub wiki at the following URL:
https://github.com/PartnerRobotChallengeVirtual/human-navigation-unity/wiki/SystemOverview

4.4. Time Limit

- The time limit for each session is N minutes ($3 \leqq N \leqq 5$).
- The task is consists of M sessions ($M \leqq 30$).
- N and M will be announced at least 7 days before the competition.
- The timer starts when the start button is clicked by the technical committee.

4.5. Score (for each session)

Action, Penalty, and Time bonus	Score
The avatar grasps the target object	*20*
Time bonus for the grasping of the target object	$\frac{remaining\ time}{time\_limit} \times 60$
The avatar grasps a wrong object (once for each object instance)	-5
The avatar places the target object in/on the destination	*20*
Time bonus for placing the target object in/on the destination	$\frac{remaining\ time}{time\_limit} \times 60$
A penalty for each optional instruction that exceeds the max. number of instructions	-3
The robot comes within α [m] of the target object (unique penalty for each target)	*-40*
The robot comes within α [m] of the destination	*-40*
The robot collides with objects except the avatar (each time)	*-10*
Total score (excluding penalties and bonuses)	***40***

Time bonuses will be calculated by two parameters *time_limit* and *remaining_time*. The *time_limit* is equal to *N*. *remainin_time* is time remaining when the avatar grasps or places the target object.

※ Test subjects will be told the scores, the penalties, and the time bonuses in advance.

4.6. The layout of Virtual Environment

- Several example environments are available from GitHub.
- Objects used in the competition include the target candidates, non-related objects, and furniture. They are provided in the Unity project files at least 14 days before the competition.
- The room layout information will be provided in Unity project files. The number of layout types will depend on the number of sessions (≦*M*). The system sends a room layout ID when the session starts. Room layouts and layout IDs will be announced at least 24 hours before the competition.
- A layout always contains only one big room not separated by walls or doors.
- The preparation of multiple types of environments is planned.
  - The environment for the competition includes furniture not provided in the sample, and changes may also be made to, for example, the quantity of furniture.

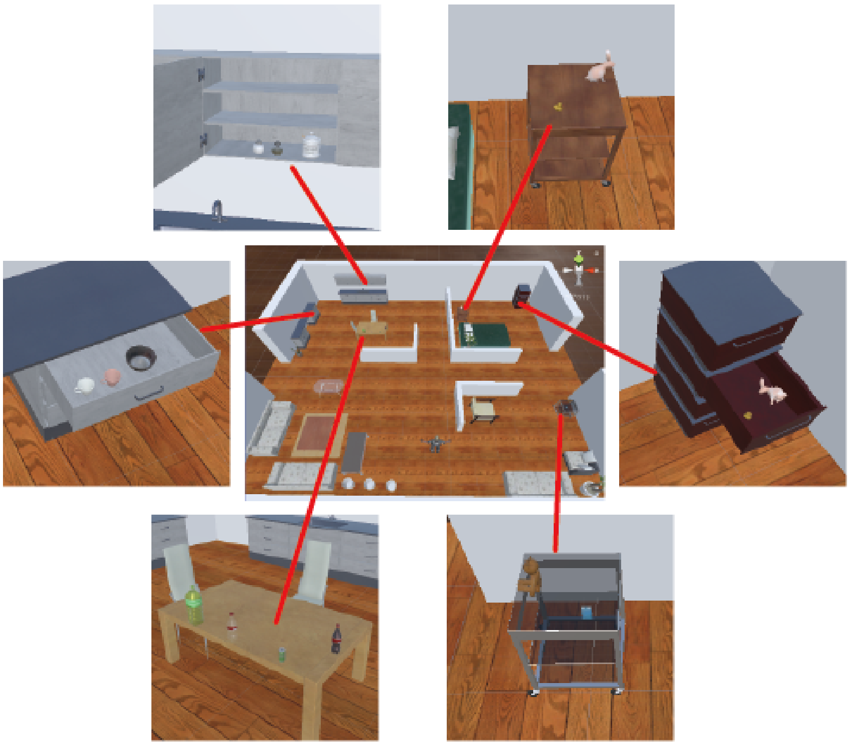

Figure 4.2: An example virtual environment for Human Navigation

4.7. Remarks

4.7.1. Objects to manipulate

- The 3D models and prefab names of the graspable objects will be released in advance.
- The position, quantity, and type of the objects changes in each session.
- Identical objects may also be placed in the environment.
- The initial position of each object is sent to the robot at the start of the task using software API. However, this message does not include information such as the piece of furniture on which the object is located.
- The object names used in the software communication correspond to the 'prefab name' information shown in the object list, which will be announced before the competition.

4.7.2. Initial position/orientation of the robot

- Initial position and orientation of the robot are (x, y, z, theta) = (0, 0, 0, 0) in all sessions.

4.7.3. Software API to refer to task information

- The team's software can refer the following task information by using a software API.
    - The Room layout ID
    - The prefab name, position, and orientation of the target object
    - The position, orientation, and size of the designated area in which to place the target object.
    - A list of prefab names, positions, and orientations of all existing graspable objects except the target object and furniture.
    - A list of prefab names and positions of furniture objects.

- The communication protocols are defined in detail in the GitHub wiki

4.7.4. Instructions from the robot

- The robot can generate natural language instructions at any time.
- Natural language instructions are provided to the test subject verbally (using SAPI) and visually (using the message board effect in Unity).
- The team can use two message boards, one is displayed above the robot and another is displayed in front of the avatar.
- Instruction sentences are shown on those virtual message boards for several seconds. The team can choose whether each message board should be used or not.
- The number of characters used in the instruction should not exceed 400.
- The number of instructions from the robot is limited to 15. If the limited is exceeded, a -3 point penalty will be given for each additional instruction.
- The robot can use gesture instructions to point to the target destination; however, when the robot comes within α [m] ($1 \leqq \alpha \leqq 5$) of the target object/destination, a -40 point penalty will be given. The α will be announce at least 7 days before the competition.
- The avatar and graspable objects never collide with the robot; therefore, there will be no penalty due to the robot colliding with the avatar and graspable objects. If the robot collides with static objects (e.g., furniture), a -10 point penalty will be given.
- The test subject can request the natural language instructions at any time by using a button on the Oculus Touch. If the robot replies to this request, the reply will be counted as an instruction. The robot also can ignore this request.
- During the practice, the test subjects will learn not to press the Oculus button, which leads suspending the task and showing the Oculus Home. Additionally, the Oculus software configuration will be set not to activate the Oculus button except long press to prevent misoperations. In case the test subject opened the Oculus Home scene, he/she will return to the session by pressing the Oculus button as soon as possible. To do so, the test subjects will also learn how to return from the Oculus Home scene to the session during the practice.
- If the test subject could not return to the session in a short time (e.g., 5 seconds) after the Oculus Home was shown, the team can request retry. In this case, the new score will be adopted even if the new score was lower than the first score. Of course, a new test subject who has no knowledge about the room layout and does not have the conflict of interest will be selected.

4.7.5. Test Subjects who log in to virtual avatar

- The test subject cannot ask any questions to the robot.
- The test subjects are selected by the executive committee. The test subjects do not have conflicts of interest in the competition.
- The test subjects must know nothing about the environment.
- Test subjects will learn the operational procedures to move and grasp objects, to open and close doors and drawers, and to request the instruction to the robot in a test environment in advance.
- The test subjects will be positioned at a certain place where they cannot look over the environment.
- Test subjects will be told the scores, penalties, and time bonuses in advance.
- The mapping between the names and appearances will be shown to test subjects beforehand. However, teams should not expect the test subjects to always remember the names. It is not the organizers' fault if the test subject does not understand the names. Teams are responsible for

ensuring the test subjects understand the sentences including objects and furniture.

4.7.6. Software API to refer to ongoing status of avatar and objects

- The robot can refer to the following information about the avatar without any penalties:
  ○ Position and orientation of the avatar's head, body, and both hands
  ○ Prefab name of the grasped object and whether the grasped object is the target object
- The robot can refer to the following information about the graspable objects without any penalties:
  ○ Position and orientation of all existing graspable objects except handles of drawer and door

4.7.7. VR devices

- The test subjects use the following devices to operate the avatar:
  - Oculus Rift & Touch
  - Wired earphones
  - Earmuffs for soundproofing

4.7.8. Voice guidance for test subjects

- The Windows PC synthesizes the voice saying of the instruction sentence using by SAPI. The test subjects always wear earphones to hear the synthesized voice.
  - The test subject cannot hear synthesized voices from other teams due to wearing earmuffs.
- SAPI is executed on the computer prepared by the competition committee.
- If a synthesized voice has not finished speaking when a new instruction message is sent from the robot, the previous synthesized voice will be canceled and the new synthesized voice will be played.

4.8. Main Window

An example of the main window of Human Navigation is shown in Fig.4.3.

Figure 4.3: Example screenshot of Human navigation

4.9. System Setup & Protocol of Competition

4.9.1. Schedule in the preparation phase

- at least 14 days before the competition: Object list in will be made public in Unity project files.
- at least 24 hours before the competition: Layout will be made public in Unity project files.
- n hours before the competition: Teams set up the robot controller PC and stop modifying its software
- The value n will be announced at least 7 days before the competition. The range of n will be 0 ≦ n ≦ 3.

4.9.2. Computer settings

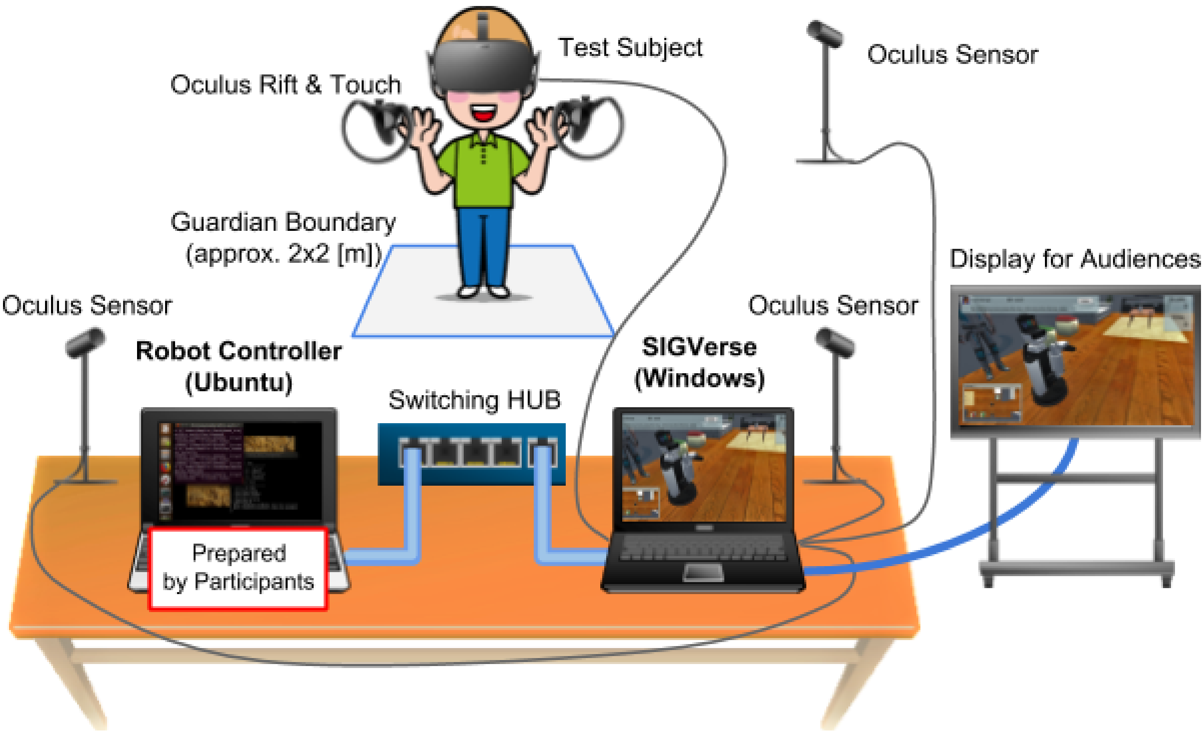

Figure 4.4: System overview for Human Navigation

4.9.3. How to start task

(1) The team should execute the robot controller at the scheduled start time of the task.
(2) The technical committee executes the Human Navigation program.
(3) The assigned test subject wears the head mounted display (HMD) and VR controllers.
(4) The technical committee presses the start button at the scheduled start time.

4.9.4. Competition procedure

- Multiple teams will compete simultaneously so that test subjects will not be able to hear instructions in advance by listening to other teams.
- The organizer starts the timer when the test subject is ready to execute the session. The start message will also be sent to the team's software module.

5. Final Demonstration

5.1 Background and Concept

The purpose of the final demonstration is to enable the teams to demonstrate their robot systems' best abilities as a partner robot to support humans without any rule restrictions. The teams are encouraged to demonstrate their latest research and developments of new approaches and applications for assistance robots in an interesting setting. The team can use any kinds of systems and devices including real robot systems; however, the main focus of the final demonstration is to utilize the virtual space towards realizing intelligent service robots. For example, the intelligent cyber-physical system will be one of the important keywords.

The team should use at least one robot simulator or piece of VR software. The use of the SIGVerse system is encouraged but not mandatory, and the type of the simulator/VR software does not affect the evaluation results.

5.2 Presentation time and schedule

- Each team has 15 minutes for the presentation including question and answers.

- 10 min
  - Preparations for demonstration presentation
  - Demonstration & Presentation
- 5 min
  - Question and answers
  - Restoring the presentation field to its original state.

5.3 Presentation Style

- Both the speech and slides must be in English.
- The team can use two PCs for the presentation. Both screens can be displayed to the audience.
- The team can use music in the presentation or video file in the slide; however, the team cannot use music protected by copyright, because the presentation will be broadcast via YouTube.
- Interviewers (evaluation committee members) might use Japanese for the Q&A. If they do, a professional translator will translate the Japanese questions into English.
- The team should use English even if the interviewers use Japanese.
- The team can use the field of the Partner Robot Challenge (Real Space) for their demonstration. In that case, the team should declare to use the field at least 1 day before the demonstration.

5.4 Interviewers

- The organizers choose the interviewers.
- Some interviewers are experts in robotics research; others are non-specialists.
- Some interviewers might not be native English speakers.

5.5 Evaluation Criteria

- The interviewer will evaluate the presentation based on the following criteria:
  - Creativity/presentation of the story
  - Effectiveness of interaction between the person and robot
  - Diversity/universality of system integration
  - Difficulty/completeness of performance
  - Relevance/practicality in daily life
- Other criteria might be added. In that case, the organizer announces the new criteria at least 7 days before the competition date.