

Interaction Debugging: an Integral Approach to Analyze Human-Robot Interaction

Tijn Kooijmans^{1,2} Takayuki Kanda¹ Christoph Bartneck² Hiroshi Ishiguro^{1,3} Norihiro Hagita¹

¹ATR Intelligent Robotics and
Communication Laboratories
Kyoto 619-0288
Japan

²Eindhoven University of Technology
Department of Industrial Design
Den Dolech 2, 5612 MB Eindhoven
The Netherlands

³Osaka University
Dep. of Adaptive Machine Systems
Osaka 565-0871
Japan

tijn@kooijmans.nu, kanda@atr.jp, christoph@bartneck.de, ishiguro@atr.jp, hagita@atr.jp

ABSTRACT

Along with the development of interactive robots, controlled experiments and field trials are regularly conducted to stage human-robot interaction. Experience in this field has shown that analyzing human-robot interaction fosters the development of improved systems and the generation of new knowledge. In this paper, we present the interaction debugging approach, which is an integral way of analyzing human-robot interaction. It is based on the collection and analysis of data from robots and their environment. Considering the current state of robotic technology, often only audio and video are insufficient for detailed analysis of human-robot interaction. Therefore, in our analysis we combine multiple types of data including audio, video, sensor values, and intermediate variables. An important aspect of the interaction debugging approach is using a tool called Interaction Debugger to analyze data. By supporting user-friendly data presentation, annotation and navigation, Interaction Debugger enables fine-grained analyses of human-robot interaction. The main goal of this paper is to address how an integral approach to the analysis of human-robot interaction can be adopted. This is demonstrated by three case studies.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems –*Evaluation/methodology, Video*

J.4 [Social and Behavioral Sciences]: *Psychology*

General Terms

Design, Measurement, Human Factors, Experimentation

Keywords

Integral approach, data analysis tool, case studies, interaction debugger, human-robot interaction, communication robots

1. INTRODUCTION

The rapidly growing interest in the field of human-robot interaction has led to major improvements in the way current robotic technology is being developed and improved. Moreover, interaction between humans and robots in general has made great

leaps since the development of interactive humanoid robots, such as Honda's Asimo [1], Sony's Aibo [2], and ATR's Robovie [3]. As the complexity and performance of robotic behavior continues to increase, such interaction with people is also becoming richer and more meaningful. To maintain this momentum, robot developers must keep focusing on strategies that improve human-robot interaction.

1.1 Analyzing human-robot interaction

To effectively improve a system, a thorough analysis of its performance must be done and used as input for further development. In the field of human-robot interaction, analyzing the behavior of both people and robots has become relevant to find opportunities for improving a robot's behavior. Lab experiments as well as field trials are frequently conducted for this purpose [4][5][6]. After an experiment or trial, one analyzes a set of audio and video data to evaluate the robot's interaction with people [4]. Generic tools originally developed for psychologists and linguists aid such analysis with annotation functionality [7][8]. A limitation of this approach is that only audio and video are analyzed.

In cases of advanced interactive robots, experiments or field trials can become complex, influenced by many factors that cannot only be conveyed by audio and video. For example, they don't show robotic intention if a behavior failed. For such analysis, one needs to consider the internal software of a robot, which can output its active behavior states. Another limitation of video is that camera views are easily blocked, for example, complicating the analysis of body contact. In the latter case, incorporating data from a robot's touch and motion sensors in the analysis is a solution. From now on, we will refer to this as integral analysis, which involves multiple types of data such as audio, video, sensors values, and internal robot variables. Early examples show such an approach based on recording and visualizing body movements [9] or gazes [10] of humans and robots while interacting. However, the application domains of these examples are limited since only one data type is available in addition to audio and video.

1.2 Interaction debugging

Interaction debugging is an integral approach for the analysis of human-robot interaction whose aim is to provide robot developers



Figure 1. Field trial with Robovie

with a tool for evaluating or debugging robotic behavior. Furthermore, psychologists can perhaps adopt the approach to analyze human responses to a robot and evaluate/debug such behavior accordingly. Essentially both seek to improve human-robot interaction.

In our approach, we emphasize the collection and analysis of multiple types of data including audio, video, sensor values, and intermediate variables. For this analysis, we have developed a software tool named Interaction Debugger that aggregates the data and presents it comprehensibly using graphical representations. Furthermore, it provides functionality to make annotations about interaction events. This combination enables effective analysis of human as well as robotic behavior. We define an analysis as effective if it leads to adjustments of the robot that improve its interaction with people.

2. DATA COLLECTION

The first step in interaction debugging is the collection of data during controlled experiments or field trials with interactive robots. It is essential to consider which types of data are necessary to collect for later analysis, which obviously depends on the emphasis of one's analysis. In this section, we describe an example setup for collecting data during field trials with interactive robots.

As an example of applying the interaction debugging approach, we studied the interaction between humans and interactive humanoid robots Robovie and Robovie-M. Robovie is a communication robot that autonomously interacts with people by speaking and gesturing (see Figure 1) [3]. Robovie-M is a small version of Robovie that can show autonomous behavior, but has no integrated sensing capability. The aim of this study was both to debug the robots' behavior as well as to collect empirical data about the behavior of people toward humanoid robots.

2.1 Data Storage

During field trials, we collected data from multiple sources: the robots and capturing PCs placed in their environment. Figure 2 illustrates the data management within this setup. Basically, all the captured data are sent to a central place to be stored, which simplifies later data retrieval. In general, data consists of a timestamp and a set of values; the format depends on the type of

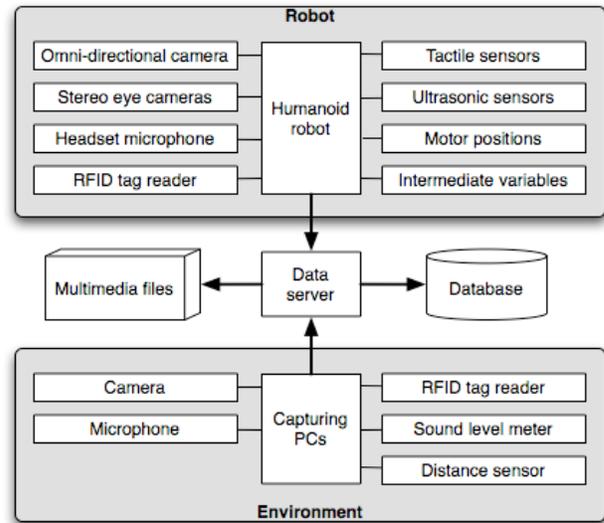


Figure 2. Data collection setup

data. For example, the data format for a sound level meter is a value between 0 and 120 decibels. For audio or video, the actual media contents are stored in the file system, and only a filename for reference is stored in the database.

To incorporate data from multiple sources, time is an important index to retrieve data later. For this reason, we use Network Time Protocol (NTP) on all the systems that collect data to synchronize clocks with an accuracy of 10 ms.

2.2 Data Types

The following data types were collected during field trials. Figure 2 shows an overview of all these types.

- Audio data captured by microphones connected to a robot or capturing PC stored in consecutive parts, typically one minute in length to limit file size and to maintain well-organized data collection.
- Video data captured by cameras in the robot and the environment. Robovie has two eye cameras and one omnidirectional camera above its head. The capturing PCs can connect multiple cameras. We used standard MJPEG compression for captured video because it allows forward and reverse play at various speeds with audio, step-play with audio, and forward and reverse frame steps. Video parts were also recorded at lengths of one minute.
- Sensor data collected from the environment come from Radio Frequency Identification (RFID) tag readers, sound level meters, and distance sensors. All these sensors can indirectly provide information about people in the environment. RFID tag readers identify people, sound level meters provide information about the noise they produce, and distance sensors show their location.
- From the robot, we collect sensor data from RFID tag readers and tactile, ultrasonic, and rotation sensors. RFID tag readers identify the people interacting with the robot, tactile sensors show where the robot was touched, ultrasonic sensors show the distance of people or objects around the robot, and rotation sensors show the position of the robot's limbs.

- Several types of output from the robot's software are collected, including behavior states, face detection results, speech detection results, emotion recognition results, speech activity, battery levels, and geographic position data. We refer to them as intermediate variables.

3. DATA ANALYSIS

To analyze data from field trials, we use the Interaction Debugger software, which consists of four important functionalities relevant for data analysis: data retrieval, presentation, annotation, and navigation. In the following subsections, we describe these functionalities in detail.

3.1 Data retrieval

We defined two modes of operation for Interaction Debugger that use different data retrieval methods: recorded and real-time. Recorded mode is intended for detailed data analysis after an experiment or trial; real-time mode is especially useful for instant optimization or debugging of a robot's behavior.

In recorded mode, data are retrieved through a database connection that contains all the data collected during an

experiment or trial. For audio or video data, the actual contents are retrieved from a local or networked file system. Settings are provided in Interaction Debugger to specify the file and database locations.

Real-time mode is a version of Interaction Debugger that immediately presents data at the event time. In that case, a direct network connection with the capturing PCs and the robots facilitates data retrieval. Real-time audio and video presentation have not yet been implemented in Interaction Debugger but will be a valuable future improvement.

3.2 Data presentation

One core functionality of Interaction Debugger is presenting data comprehensibly using visualizations tailored to the data. When using the software, one can open windows for every data type. Audio and video windows are loaded by selecting the audio or video source in the multimedia window (Figure 3.6). Other data types are accessible from the menu bar of Interaction Debugger's main window. Figure 3 shows a number of example windows that can be loaded:

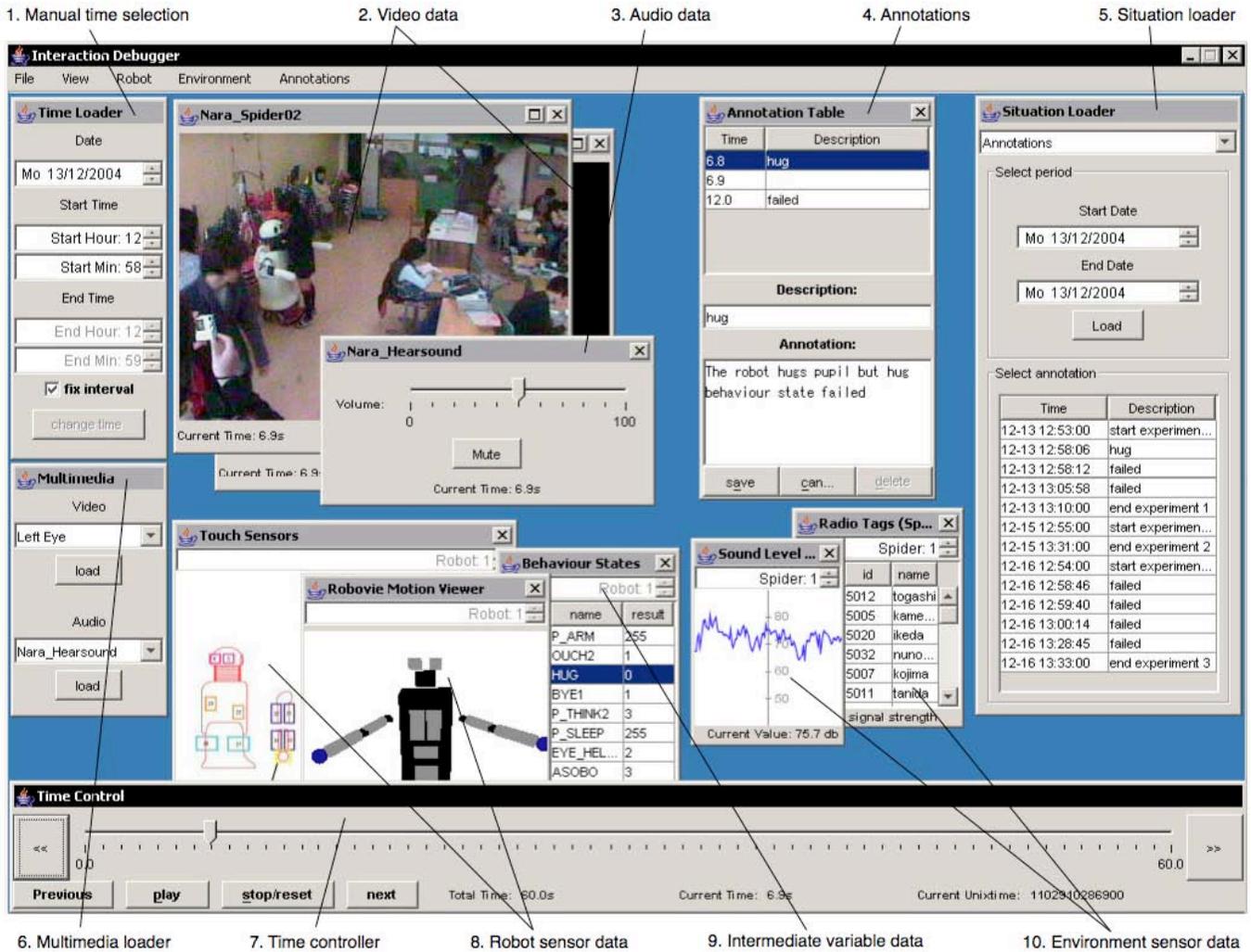


Figure 3. User interface of Interaction Debugger



Figure 4. Video data from multiple camera sources: environment, robot eye, and omnidirectional cameras

- Figures 3.2 and 3.3: Video and audio windows that can be simultaneously loaded from multiple sources (see Figure 4).
- Figure 3.8: Visualizations of the robot's touch and motion sensors. The former start blinking when the robot is touched. The latter are visualized by a three-dimensional model of the robot that resembles its motion. Other available windows for robot sensor data present data from ultrasonic distance sensors (see Figure 5) and RFID tag readers.
- Figure 3.9: Active behavior states of the robot, which is one of its intermediate variables. The result of each behavior state is displayed, indicating success or failure.
- Figure 3.10: Environment sound level and a list of people in the robot's environment. The latter is based on RFID tag readings. Another available window for environment sensor data present data from distance sensors (see Figure 7, Step 1).

Our implementation of Interaction Debugger incorporates data presentation windows optimized for Robovie. To increase comprehensibility, we present the robot's sensor data onto graphical representations of Robovie, as demonstrated in Figure 5. Apart from that, Interaction Debugger features standard presentation styles including tables and line charts. For such textual data as behavior states and RFID tag readings, we use tables (Figures 3.9 and 3.10). For such single sensor values as sound levels we use line charts.

Since Interaction Debugger might be used for different types of robots in other situations, it has been designed with modular software architecture. The graphical presentation of data and their underlying management have been clearly separated, making it easy for developers to modify or design new presentation styles. Moreover, it enables easy implementation of new data types.

3.3 Data annotation

To aid the analysis of human-robot interactions, a special annotation window (Figure 3.4) has been incorporated in Interaction Debugger. Inspired by existing audio and video annotation software [7][8], this feature allows users to describe

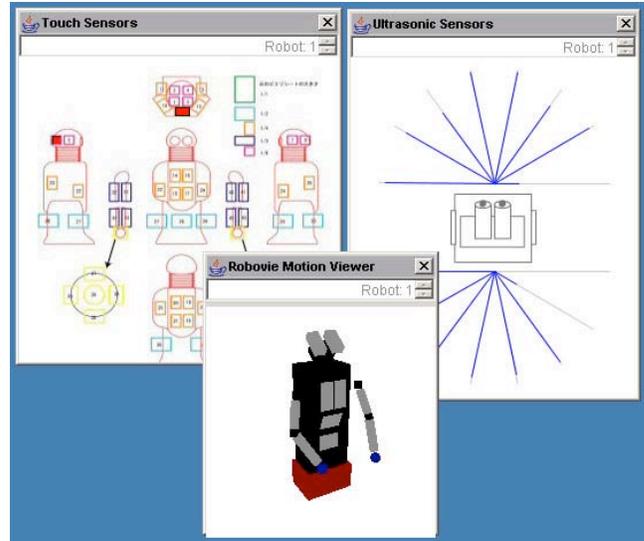


Figure 5. Graphical representations of robot sensors

every frame of the data collection. For example, researchers could use this functionality to make detailed descriptions of human behavior. Moreover, it could be a useful data navigation method, as explained in the next subsection.

3.4 Data navigation

Navigating data in Interaction Debugger is based on the selection of a time interval, consisting of a start and an end time. The user-interface supports three ways of selecting a time interval, which are each useful in different situations. One can choose manual selection by specifying a date, a start time, and an end time in the time selection panel (Figure 3.1). However, in many cases where interesting data are available, it is desirable to select a specific event. For this purpose, one can use a situation loader (Figure 3.5) to recall the time of an annotation previously recorded.

An additional feature that we implemented as part of the situation loader is behavior-based situation loading that enables users to retrieve a list of all the events where certain robotic behavior states were active and to load time intervals accordingly. For this feature, the robot must support the output of behavior states.

After a time interval has been loaded, a timeline window is activated at the bottom of the screen (Figure 3.7) that enables time control of all the data presentation windows currently loaded. To browse through data, one can manually move the timeline or use the play function.

4. CASE STUDIES

To demonstrate the interaction debugging approach, we present three case studies in which the approach was adopted. For each one, a step-by-step description will clarify how Interaction Debugger was employed for data analysis. In the case studies, the data analyzed were collected with the example setup discussed in section 2.

To improve Robovie's interaction with people, field trials are regularly conducted to afford people having interaction with the robot. During these trials, interaction is analyzed, and adjustments to the robots are made to improve its behavior. The case studies



Figure 6. Field trial with Robovie-M

we present in this section were conducted for this purpose. Because of the realistic situations, they provide a good illustration of our approach's practical applicability.

4.1 Optimizing thresholds for a robot

During a field trial, a number of robots were placed in the Osaka Science Museum to interact with people. This particular setting is part of our previous research activities [11]. Since Robovie-M was programmed to explain exhibits to visitors for this trial (see Figure 7), it had to detect the presence of people and proactively draw their attention. Because Robovie-M has no integrated sensing capability, several sensors were placed around it to enable presence detection. For example, an infrared sensor was placed under the robot to measure the distance of objects in the environment, and a sound level meter distinguished background noise from human speech. To use these sensors for presence detection, they are read by Robovie-M's control software and interpreted based on thresholds. Because every environmental situation is different, these thresholds have to be set manually.

The real-time mode of Interaction Debugger was employed in this situation to optimize the presence detection thresholds. The robot developer used the following method (see Figure 7):

- Step 1: Using the visualization of the infrared sensor, for

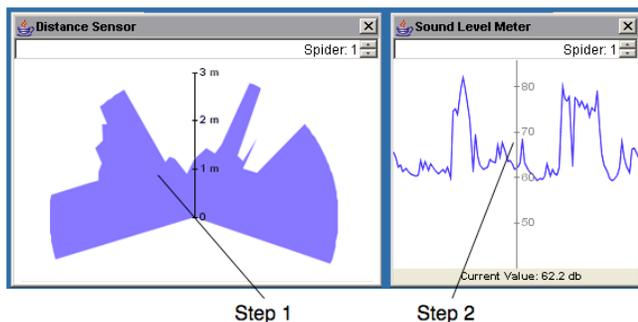


Figure 7. Screenshot case study 1

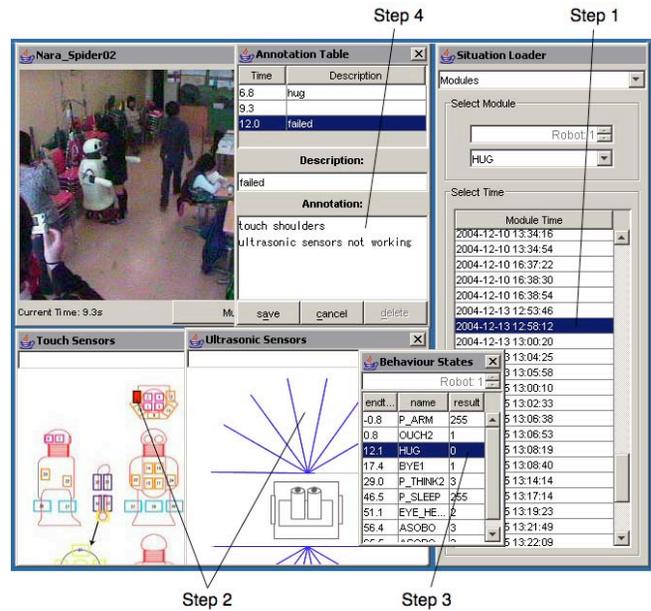


Figure 8. Screenshot case study 2

each occasion he analyzed the distance at which people approached the robot and the angles at which they started to interact with it. He used this information to set the corresponding thresholds.

- Step 2: Sound level meter visualization was analyzed to set the voice detection threshold of the robot. In the picture, peaks represent human speech.

Using Interaction Debugger, the developer could successfully optimize the presence detection mechanism in a relatively short time span.

4.2 Debugging a robot's behavior

During a field trial at a Japanese elementary school, Robovie was positioned in a classroom for eighteen days. The goal of the experiment was to study the social interaction and the establishment of relationships between pupils and the robot. This particular setting was also used for our previous research activities [5].

Robovie is designed to sometimes exhibit hugging behavior during interaction with people if they keep reacting to it. However, hugging didn't always appear successful. In this experiment, a hug was considered successful if the robot closed its arms around the user when he or she stepped toward the robot with open arms.

A robot developer used Interaction Debugger to analyze the data recorded during three days of trials to debug the hugging behavior of the robot. His method can be summarized as follows (see Figure 8):

- Step 1: In the behavior-based situation loader, the user selected hug behavior to retrieve a list of all hugging events. By double clicking on an event, the corresponding time is automatically loaded. The data windows were loaded, as described in section 3.2.
- Step 2: For each event, he analyzed the touch sensor and distance sensor conditions of the robot that activated hugging behavior. A highlighted touch sensor shows that it was

pressed; when an object is near the robot, the blue lines of the ultrasonic sensor display are interrupted. The time controller (Figure 3.7) of Interaction Debugger enabled him to study the data frame by frame.

- Step 3: For each event, he also checked the success of the hugging behavior by reading the result values of the behavior state. Robovie outputs results for every finished behavior state, which are displayed in the table. For hugging, a zero value means that the behavior was interrupted or was not finished successfully.
- Step 4: Finally, he annotated for each event the success of the hug and the corresponding sensor conditions.

Twenty percent of the hugs were not successful. The ultrasonic sensor window revealed the failure of the robot to detect objects in front it during all unsuccessful hugs. This instability of the ultrasonic sensors indicates the cause of the problem. With this information, the developer debugged the robot and improved its hugging behavior. Although this is a simple case of debugging, we can consider it an effective analysis.

4.3 Studying human behavior

For the same field trial at the Osaka Science Museum as in case study 1, a researcher with a background in cognitive psychology carried out an empirical study on the behavior of people who interacted with Robovie-M. His goal was to learn how the crowd around a robot influences the way people react to it. This knowledge might be useful later to improve the way a robot initiates interaction with people in different crowds.

For analyzing human behavior, he adopted an observation technique established in psychology based on analyzing data by making annotations using a code protocol. In this case, he considered every event where someone interacts with the robot and coded human actions following a set of parameters that included such personal information as adult/child, alone/group and gender, and information about the interaction such as type of behavior, cause of behavior, distance from robot, and crowdedness. A comparable example of such a coding system is the Facial Action Coding System (FACS) developed by Ekman et al [12]. Today, FACS is widely used for facial emotion recognition. Coding data by hand enables the analyst to use an exploratory approach and have quantifiable results at the same time.

The factors that played an important role in his study were the positions of people and the environmental sound level. Both provide information about crowdedness. He used Interaction Debugger as a tool to analyze how these factors influence human behavior toward the robot by carrying out the following method (see Figure 9):

- Step 1: The first step was to find events where people approached the robot. To achieve this, the user moved the timeline slider to rapidly skip through the scenes. Whenever noticing people approaching the robot, he used the play function to scrutinize the interaction event.
- Step 2: For each interaction event he coded the start and end time of the event, the person's ID (e.g., B156), adult/child, alone/group, and gender in the annotation window.
- Step 3: When analyzing the video, he checked every human action for behavior that matched one of the predefined behaviors considered interesting (e.g., moving, speaking,

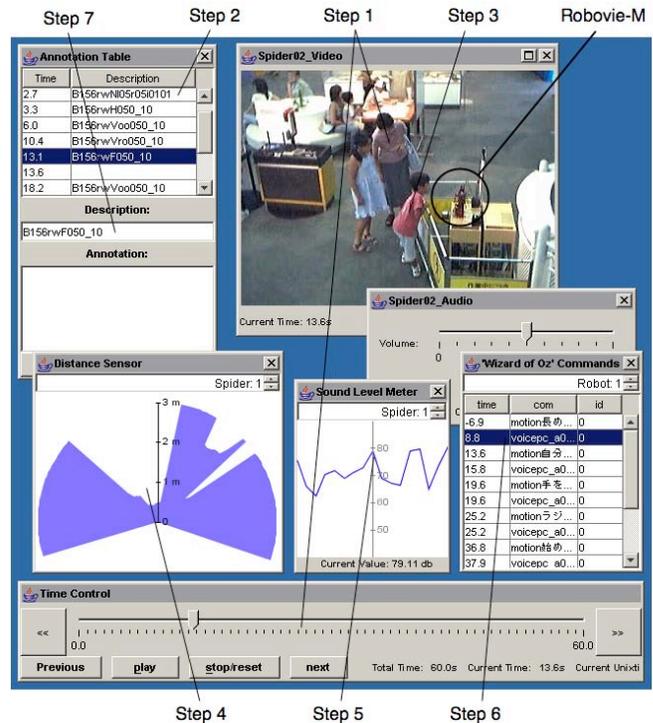


Figure 9. Screenshot case: study 3

imitation, waving, bending, touching, etc.). To consider the human behavior in detail, he viewed the data frame by frame using the time controller.

- Step 4: To analyze the distance and position of people from the robot, he observed the distance sensor window while navigating frame by frame. For every human action, he measured the distance between human and robot.
- Step 5: He analyzed the sound level of the robot's environment by checking the decibel value in the sound level meter window. For every human action, he calculated the average sound level as an indicator for crowdedness.
- Step 6: To analyze the cause of human behavior, he checked the type of behavior the robot performed in the "Wizard of Oz" command window, which shows the active behavior states for Robovie-M.
- Step 7: For each interaction event, he recorded every human action and the results from steps 2, 3, and 4 as coded annotation.
- Step 8: After using Interaction Debugger, he used spreadsheet software to interpret the codes and compute the statistical results. Example computations include the interaction time of people relative to distance from the robot or interaction time relative to crowdedness.

His analysis revealed that people showed different patterns of approaching the robot in different crowd situations. This knowledge can later be used to make the robot automatically infer that people are interested in it by measuring crowdedness and movement of people. The results of this study will be expanded in a future paper [13].

5. DISCUSSIONS

5.1 Contributions

This paper presented an integral approach to analyze human-robot interaction, which we believe is an essential part in the development process of interactive robots. By adopting this approach, robot developers can efficiently improve robot interactivity. Improving hugging behavior is a simple example, but more complex situations in which an integral approach could help are easily imaginable: for instance, the evaluation of speech recognition by analyzing audio data, background noise level, and intermediate variables that indicate recognized speech.

For psychologists, the interaction debugging approach is useful to aid qualitative data analysis techniques, such as the observation method, which is often adopted in human behavior analysis. Another case in which interaction debugging could have been useful was the development of a human friendship estimation model for communication robots [14]. In this research, inter-human interaction was analyzed in the presence of a humanoid robot.

Psychologists can evaluate human responses to robotic behavior by studying human-robot interaction, which can help robot developers to adjust the robot and optimize its behavior. We believe such an interdisciplinary approach is essential for improving human-robot interaction.

5.2 Evaluation of the approach

Unlike the evaluation of a method, it is difficult to evaluate a new methodology. Since no related methodology was available for comparison, we did not conduct a controlled experiment to evaluate the interaction debugging approach as a whole.

The interaction debugging approach can be decomposed in the following methods: “showing sensory information” and “integration of multi-modal information.” To evaluate the effectiveness of the first method, for instance, we could conduct a controlled experiment that compares analysis results with and without the sensory information or intermediate variables. However, as shown in the case studies, we often cannot accomplish the analysis goal without this information.

To test the integration of the multi-modal information we could compare the use of Interaction Debugger with a common controller of audio/video and specialized software for displaying sensory information. However, a tool that integrates these components is obviously more effective.

We feel that individual evaluation of both methods doesn't lead to a clear impression about the validity of the interaction debugging approach as a whole. Therefore, instead of conducting such experiments, we focused on the introduction of our integral approach by presenting case studies in this paper.

5.3 Evaluation of software

Since Interaction Debugger is intended for people from different disciplines who do not have the same experience with the technical aspects of robotics, we consider usability a key evaluation point. Based on the usability goals specified by Preece et al. [15], “time to learn” and “retention over time” were selected as the main criteria for optimizing the user-interface.

Within the process of optimizing the usability of Interaction Debugger, the first method employed was expert reviewing, which is commonly used in software development to evaluate a

user-interface by determining conformance with a short list of design heuristics. We used Shneiderman's “eight golden rules of interface design” [16]. Furthermore, the case studies were part of a user-centered method to optimize Interaction Debugger's user-interface [17]. For each case study, user interaction with the software was studied, and feedback was requested to generate usability improvements.

From our observations of people who used Interaction Debugger we drew some conclusions that illustrate its current usability performance. The simple structure of its user-interface made it easy for people to start working with it. If experienced with window-based GUIs, new users only needed a brief explanation of the different windows to get started. For non-novice users, the software provides enough shortcuts to efficiently control the user-interface. Examples include mouse scrolling to control time and key combinations for adding annotations.

A problem worth mentioning concerns the organization of windows in Interaction Debugger. Because the amount of windows can become large for certain analysis tasks, having a friendly way of positioning them on the screen is helpful. We decided to let the user manage the organization, which means that the software remembers the last position of a window. This enables users to personalize the software to create a comfortable working environment.

Another problem we encountered during the development of Interaction Debugger is the synchronization of data, which is critical for accurate analysis. Although the data clocks of the capturing computers are synchronized by NTP, the software that records data often causes delay. We implemented a manual delay compensation function in Interaction Debugger to address this concern.

5.4 Limitations

In this paper, we only demonstrated three simple case studies of the interaction debugging approach. All cases were related to independent projects and were not part of any large-scale engineering process because the software has been prepared very recently. Hence, applicability and effectiveness for large-scale development is not yet clear.

At the moment, the generalizability of our approach is still unknown. Since it was only tested in a limited number of applications we can't determine in which cases the approach will be applicable and effective and in which cases it will not be. We believe that using the interaction debugging approach in our robot development activities will foster a better view on this. Moreover, we would like to encourage other robot developers to adopt this method and contribute to this field.

Another limitation of the current status of development is related to the types of data required in integral analysis. Currently, no guidelines have been developed that give such indications. In our examples, we used robot sensors, environment sensors, and intermediate variables. However, for certain purposes one might not need all this data.

6. CONCLUSION

We believe an integral approach that analyzes human-robot interaction involving audio, video, sensor data and intermediate variable data has become necessary because of the contemporary complexity of interactive robots. To aid such data analysis, a tool named Interaction Debugger has been developed that allows us to

conduct interdisciplinary projects for investigating human-robot interaction while at the same time offering an environment that encourages collaboration between robot developers and psychologists. We demonstrated the practical applicability of the interaction debugging approach by addressing three different case studies. In all of them, the use of Interaction Debugger led to an effective analysis of human-robot interaction. However, we are aware that this only illustrates a limited number of applications and doesn't compare its effectiveness to other methodologies. The approach's novelty limited us to do so. Hence, we hope to inspire researchers to adopt comparable methods to generate more experience in this field.

7. ACKNOWLEDGMENTS

This research was supported by the Ministry of Internal Affairs and Communications of Japan. We want to thank Shogo Nabe, Yoshikazu Koide, and Jerry Lin for their valuable contributions on the development and optimization of Interaction Debugger.

8. REFERENCES

- [1] Sakagami, R., Watanabe, C., Aoyama, S., Matsunaga, N., Higaki, N., and Fujimura, K., The intelligent ASIMO: System overview and integration, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2002, pp. 2478-2483.
- [2] Fujita, M., AIBO: Toward the era of digital creatures, *Int. J. Robot. Res.*, vol. 20, no. 10, pp. 781-794, 2001.
- [3] Ishiguro, H., Ono, T., Imai, M., Maeda, T., Kanda, T., and Nakatsu, R., Robovie: an interactive humanoid robot", *Int. J. Industrial Robot*, Vol. 28, No. 6, pp.498-503, 2001.
- [4] Dautenhahn, K. and Werry, I., A Quantitative Technique for Analysing Robot-Human Interactions, in *Proc. of the 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002, pp. 1132-1138.
- [5] Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H., Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial, in *Human Computer Interaction*, Vol. 19, No. 1-2, pp. 61-84, 2004.
- [6] Siegwart, R., Arras, K.O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Pignet, R., Ramel, G., Terrien, G., and Tomatis, N., "Robox at Expo.02: A Large-Scale Installation of Personal Robots," in *Robotics and Autonomous Systems*, vol. 42, issue 3-4, pp. 203-222, 2003.
- [7] Quek, F., Shi, Y., Kirbas, C., and Wu, S., VisSTA: A Tool for Analyzing Multimodal Discourse Data, in *Seventh International Conference on Spoken Language Processing*, 2002, Denver, CO.
- [8] Kipp, M. and Anvil, A., Generic Annotation Tool for Multimodal Dialogue, in *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1367-1370, Aalborg, 2001.
- [9] Kanda, T., Ishiguro, H., Imai, M., and Ono, T., Development and Evaluation of Interactive Humanoid Robots, In *Proceedings of the IEEE*, 2004, pp. 1839-1850.
- [10] Sumi, Y., Ito, S., Matsuguchi, T., Fels, S., and Mase, K., Collaborative Capturing and Interpretation of Interactions, *Pervasive 2004 Workshop on Memory and Sharing of Experiences*, pp. 1-7, 2004.
- [11] Nomura, T., Tasaki, T., Kanda, T., Shiomi, M., Ishiguro, H., and Hagita, N., Questionnaire-Based Research on Opinions of Visitors for Communication Robots at an Exhibition in Japan, *International Conference on Human-Computer Interaction (Interact 2005)*, 2005.
- [12] Ekman, P. and Friesen, W. V., *Manual for the Facial Action Coding System and Action Unit Photographs*, Consulting Psychologists Press, 1978.
- [13] Nabe, S., Kanda, T., Hiraki, K., Ishiguro, H., Kogure, K., and Hagita, N., Analysis of Human Behavior in an Open Field to Improve Communication Robots, *Human-Robot Interaction Conference 2006*, (under review).
- [14] Nabe, S., Kanda, T., Hiraki, K., Ishiguro, H., and Hagita, N., Human Friendship Estimation Model for Communication Robots, 2005 IEEE-RAS International Conference on Humanoid Robots, December 2005 (to appear), available at <http://www.irc.atr.jp/~kanda/pdf/nabe-humanoids2005.pdf>.
- [15] Preece, J. Rogers, Y., and Sharp, H., "Interaction Design: Beyond Human-Computer Interaction," John Wiley & Sons Inc., New York, chapter 1, 2002.
- [16] Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Longman, Inc, MA, 1998.
- [17] Norman, D. and Draper, S., *User Centered System Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Inc. Mahwah, NJ, 1986