Accelerating Robot Development Through Integral Analysis of Human–Robot Interaction

Tijn Kooijmans, Takayuki Kanda, *Member, IEEE*, Christoph Bartneck, Hiroshi Ishiguro, and Norihiro Hagita, *Senior Member, IEEE*

Abstract—**The development of interactive robots is a complicated process, involving a plethora of psychological, technical, and contextual influences. To design a robot capable of operating "intelligently" in everyday situations, one needs a profound understanding of human–robot interaction (HRI). We propose an approach based on integral analysis of multimodal data to pursue this understanding and support interdisciplinary research and development in the field of robotics. To adopt this approach, a software tool named Interaction Debugger was developed that features user-friendly navigation, browsing, searching, viewing, and annotation of data; it enables fine-grained inspection of the HRI. In four case studies, we demonstrated how our analysis approach aids the development process of interactive robots.**

Index Terms—**Data analysis, human–robot interaction (HRI), integrated approach, Interaction Debugger, multimodal data.**

I. Introduction

THE GROWING interest in the field of robotics has increased the possibilities for robots to interact with people in natural ways. Humanoid robots, including Honda's Asimo [1], Sony's QRIO [2], and ATR's Robovie [3], contribute to the shared vision of bringing robots into people's everyday lives. Whether in the role of a household assistant, social partner, or babysitter, there will always be a need for robots to interact with people and their environments. Therefore, we believe the study of human–robot interaction (HRI) deserves a central role in the robot development process.

Developing interactive robots that will communicate with people in everyday situations is nowadays clearly considered an interdisciplinary process involving psychology, cognitive science, and engineering [4]–[8]. Engineers study the HRI to develop and improve robots, while psychologists aim for a better understanding of human attitudes, roles, and expectations toward robots. The process is inherently entangled, since on the one hand, engineers require behavior frameworks developed by psychologist to help them analyze the HRI. Psychologists, on the other hand, need to be aware of the technical limitations and possibilities when developing robot behavior and creating observation frameworks. The following scenarios briefly describe how they can use data analysis for such purposes.

First, humanoid robots generally incorporate large amounts of sensors and actuators that are controlled by their artificial brains. One major engineering challenge is to process such acquired sensor information so that the robot can perform an appropriate behavior in a certain situation. By studying sensor data triggered by a user's action or environmental condition, engineers can design a robot to anticipate this information and produce an appropriate reaction.

Another scenario for studying the HRI has a more psychological-oriented motive. When people interact with a robot, one could analyze their attitudes and behavior toward it. A well-known means for conducting such a study is based on field trials. In [9]–[13], scenes of a field trial are analyzed and used for evaluation and improvement of robot behavior or design. In the long run, this could yield general knowledge about the HRI.

In both cases, data analysis is used for gaining better understanding of the HRI. Software originally developed for psychologists and linguists is available that aids this work with annotation and coding functionalities [14]–[16]. These, however, are limited to audio and video, decreasing their usefulness, especially for the first scenario. Other tools include body movements [17] or gazes [18] of humans and robots during interaction. However, the application domains of these examples are limited since only one type of data is available in addition to regular audio and video.

By integrating information from such modalities as sound, vision, object positioning, person identification, and body contact with audio and video, one can form a more complete overview of a situation, which leads to more effective analyses. We consider an analysis effective when it aids the development process of the robot or when it leads to increased knowledge about human attitudes toward robots.

Using this integrated approach one could, for instance, analyze which sensor values of a robot are triggered by certain human behavior, or if the internal states of a robot are activated in appropriate situations. From a psychological perspective, one could seek correlations between the distance from and behavior toward a robot or investigate such human attitudes as responses to body contact.

Manuscript received September 12, 2006; revised May 16, 2007. This paper was recommended for publication by Associate Editor C. Laschi and Editor H. Arai upon evaluation of the reviewer's comments. This work was supported by the Ministry of Internal Affairs and Communications of Japan.

T. Kooijmans is with Studio Sophisti, 1017 BX Amsterdam, The Netherlands (e-mail: tijn@kooijmans.nu).

T. Kanda and N. Hagita are with the Intelligent Robotics and Communication Laboratories, Advanced Telecommunications Research Institute International, Kyoto 619-0288, Japan (e-mail: kanda@atr.jp; hagita@atr.jp).

C. Bartneck is with the Department of Industrial Design, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: christoph@bartneck.de).

H. Ishiguro is with the Department of Adaptive Machine Systems, School of Engineering, Osaka University, Osaka 565-0871, Japan, and also with the Intelligent Robotics and Communication Laboratories, Advanced Telecommunications Research Institute International, Kyoto 619-0288, Japan (e-mail: ishiguro@ams.eng.osaka-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.906263

Fig. 1. Field trial with Robovie.

We developed a software tool named Interaction Debugger that helps people analyze multimodal data collected during experiments. This software enables engineers and psychologists to cooperate in the analysis of the HRI that may ultimately lead to the development of this field. Interaction Debugger promotes an integrative working process, which is necessary in the HRI field.

We previously reported a preliminary version of Interaction Debugger [19]. Currently, the software has reached a more mature level and can be applied as a full-fledged workbench to monitor, manage, and analyze experiments. New functionalities such as "data browser" and comprehensive search functions (explained in Section III) are key factors.

First, we will discuss an example setup that highlights the data types that can be included in Interaction Debugger. Next, we will describe the software's architecture, its functionality, and its workflow. Then, we present several case studies that provide an informal evaluation of the software. Last, we discuss the contributions and limitations of the software before we end with our conclusion.

II. Example Setup

Before conducting an integral analysis, one first has to gather the necessary data during experiments or field trials. It is essential to consider which modalities and types of data are necessary to collect for later analysis, which depends on the focus of one's study. In this section, we describe an example setup for gathering data during field trials with interactive robots (see Fig. 1). In this setup, we studied the interaction between humans and Robovie and Robovie-M, which are interactive humanoid robots. Robovie is a communication robot that autonomously interacts with people by speaking and gesturing [3]. Robovie-M is a small version of Robovie that can show autonomous behavior, but has no integrated sensing capabilities. Instead, we used networked sensors in the environment to feed information to the robot. The case studies in Section IV describe the goals of these experiments and the data analysis process using Interaction Debugger in detail.

For our study, in addition to regular audio and video, the following modalities of data were collected: sound, vision, person identification, motion, body contact, distance, and robotic behavior. For each modality, we recorded several types of sensory information and/or intermediate variables, which were produced by the robot's software or external sensors. Intermediate variables can provide such information about the robot's state as its active behavior, battery level, or the words it speaks. In Section III-A, we clarify the technical details of data gathering and storage. Here is a complete overview of the data types used in our Robovie study.

1) *Sound:* This contains sensor data of sound level meters in the robot environment as well as internal sensors. Two intermediate variables are related to sound: one indicates if the robot is in the talking mode and the other tells if the robot is in the listening mode.
2) *Vision:* For the eye and omnidirectional cameras used inside the robot, a diff-value is calculated that provides information about the activity in the camera sight. The robot also has a face recognition mechanism that outputs a variable the size of a recognized face and another variable that indicates if the robot's sight is blocked.
3) *Person Identification:* Throughout the field trials, we asked people to wear radio frequency identification tags (RFID). The robot as well as several places in its environment contains a RFID reader, whose software outputs tags that are in range. This gives information about the people in the robot environment and near it.
4) *Motion:* This includes the motor positions of all the robot's joints.
5) *Body Contact:* The only data type that belongs to this modality is touch sensor information, which indicates the state of each touch sensor: pressed or not.
6) *Distance:* This contains information from distance sensors in the robot and range sensors in the environment. Both are used for detecting objects (including people) around the robot.
7) *Robotic Behavior:* Robovie operates based on programmed behavior modules that are activated using rules. For example, when it detects a new person within a predefined range it activates the greeting behavior. There is also a possibility for "Wizard of Oz" control, which refers to manually activated behavior by someone remotely controlling Robovie. The activated behavior modules as well as the "Wizard of Oz" commands are recorded.

For audio and video, we used the following configuration.

1) Audio data are captured by microphones connected to a robot or capturing personal computer (PC) stored in consecutive parts, typically 1 min in length to limit file size and to maintain a well-organized data collection. Using 1-min chunks also increases performance for later data retrieval.
2) Video data are 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 to multiple cameras. We used Intel's Indeo 4.5 compressor 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 1 min.

III. Interaction Debugger

Interaction Debugger was developed to help people analyze data that were collected during experiments or field trials. We are aware that the software is an analysis tool. While it sounds unusual for psychologists to "debug" interaction, in the field of engineering, people's motivation for analysis is to debug their engineering products. Thus, we named this tool "Interaction Debugger" to entice engineers involved in the analysis of the HRI.

There are several requirements for a tool that aids an integral analysis process. We considered the following requirements when developing Interaction Debugger.

1) It should be easy to see when data are available for specific types (e.g., video, audio, touch sensor data, etc.) and sources (e.g., robot, sensor PC 1, sensor PC 2, etc.). This is especially useful when experiments are conducted over multiple days or when one wants to compare data from multiple experiments.
2) Data relevant for end-users should be easy to find, for example, quick retrieval of an overview of all events where people are around the robot.
3) Easy time navigation: quickly jumping between days, hours, and minutes. Furthermore, there should be a simple time controller that supports playback, pausing, frame stepping, and forward and backward dragging.
4) The data types should be visualized in a way that is understandable by the people involved in the analysis process.
5) A user should be able to bookmark interesting events and freely make annotations about the data.
6) Interaction Debugger should be extendible in a way that allows intermediate developers to easily add new data types.

In this section, we outline how Interaction Debugger was developed to meet these criteria from a technical point of view as well as an end-user point of view.

A. Technical Details

The software was developed using Java 1.5 to realize platform independence and to make it easily extendible (see Section III-C). We used Java Media Framework 2.1.1 to implement audio and video playback.

1) System Architecture: The Interaction Debugger software can be decomposed into multiple layers and packages to make its structure understandable. The application itself consists of a user interface layer and a layer that contains utilities that provide the underlying functionalities for the user interface (see Fig. 2). Furthermore, we consider some operating system components on which the application runs.

The application's user interface is a desktop environment that contains packages of tools, utility frames, search frames, and data frames. When discussing frames, we refer to windows inside Interaction Debugger's desktop. The data frames package consists of a series of frames that show data to an end-user from the available data types. These data frames serve the core functionality of the Interaction Debugger, displaying video, audio, sensor information, and intermediate variables. To look for interesting data, we designed several frames that provide search functions including annotations or robot behavior, filter-based searches, and a structured query language (SQL) query search for professional users. The search functions are further clarified in Section III-B. A collection of utility frames is also included, which consists of the following:

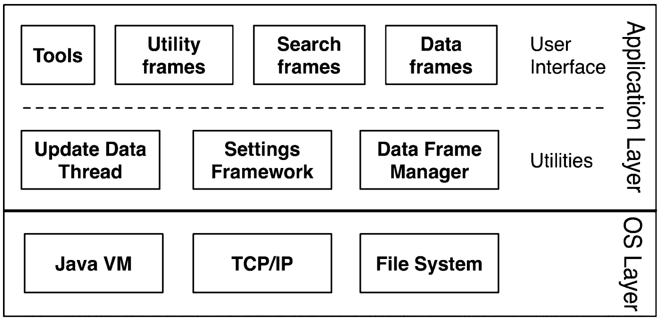


Fig. 2. Logical structure of system architecture.

1) a loading dialog that shows a progress bar when the Interaction Debugger is busy;
2) a time controller for navigation and playback of data;
3) a time browser to choose a day and time interval to analyze;
4) a time information frame that communicates the time to a user while analyzing data.

The final package of the user interface is a set of tools including:

1) frames for importing data from and exporting data to a text file;
2) a data browser to view when data are available for specific types and sources;
3) a frame that enables end-users to synchronize multiple data types. For example, if the audio and video are not exactly synchronized due to unsynchronized capturing systems, the user can set a delay for the faster one.

The utilities of the Interaction Debugger realize the following functionalities.

1) The Update Data Thread updates all the data frames when needed. This is when a new time interval is loaded, when the time controller is dragged, or, every 10 ms. when the time controller is in playback mode.
2) To maintain configurability in the Interaction Debugger, we use a settings framework, which includes functionalities to save and retrieve data from setting-files in XML format and generate a configuration panel to interface the settings to end-users. Settings include a database, file locations, table names, and settings to modify the Interaction Debugger's user interface. It is possible to create more then one settings file to serve multiple projects.
3) The Data Frame Manager is the part of the Interaction Debugger responsible for the placement of data frames on the desktop. It remembers the size and position of

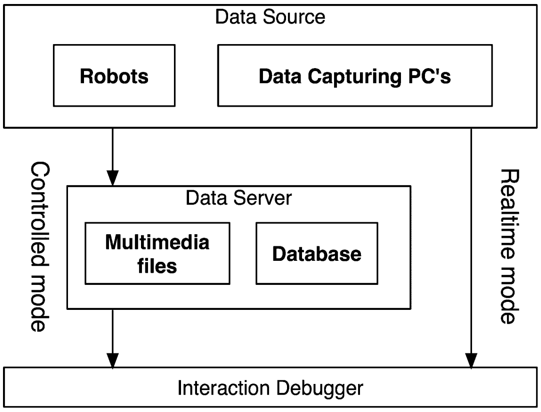


Fig. 3. Data flows.

the windows so that end-users can create personalized environments.

2) Controlled and Realtime Modes: We defined two modes of operation for the Interaction Debugger that use different methods of data retrieval: controlled and realtime (see Fig. 3). Controlled mode is intended for detailed data analysis after an experiment or trial; realtime mode is especially useful for instant optimization or debugging of a robot's behavior. In controlled mode, data are retrieved from a database 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 the Interaction Debugger to specify the file and database locations.

In realtime mode, the Interaction Debugger immediately presents data at the event time. In that case, a direct network connection with the capturing PCs and the robots facilitates data retrieval through a transmission control protocol (TCP)/Internet protocol (IP). Realtime audio and video streaming have not yet been implemented in the Interaction Debugger, but will be a valuable future improvement.

3) Data Gathering: During experiments or field trials, we collected data from multiple sources: the robots and capturing PCs placed in their environment. Fig. 3 illustrates the data flow within this setup. Basically, all 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 data. For example, the data format for a sound level meter is a value between 0 and 120 dB. For audio or video, the actual media contents are stored in the file system, and only a filename is stored in the database for reference.

Sensor and intermediate variable data were recorded at intervals of 20–200 ms, depending on the need for precision in analysis. A shorter interval means higher precision, but it also requires more processing capacity as well as more network bandwidth. For audio and video, a value is only sent to the database every 60 s when a new part is recorded.

Time is an important index when retrieving data later to incorporate them from multiple sources. For this reason, we use the network time protocol (NTP) on all the systems that collect data to synchronize clocks with an accuracy of 10 ms. If in some cases a specific data type is still not synchronized, end-users can manually adjust its timing using the data synchronizer tool in the Interaction Debugger.

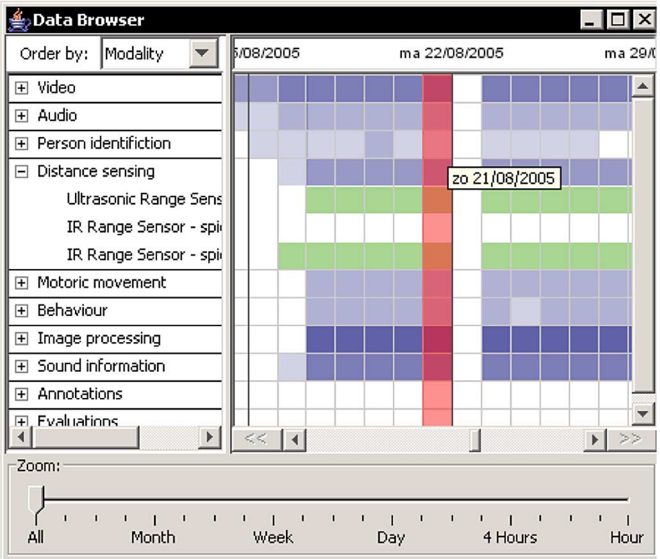
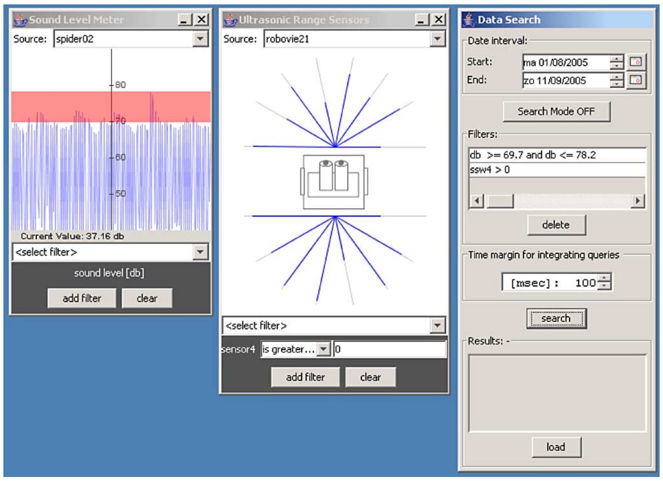


Fig. 4. Data browser.



Source: spider02

Current Value: 37.16 db

<select filter>

Source: robovie21

<select filter>

sensor4 is greater... 0

Date interval:

Start: ma 01/08/2005

End: zo 11/09/2005

Filters:

db >= 69.7 and db <= 78.2

ssw4 > 0

Time margin for integrating queries

[msec]: 100

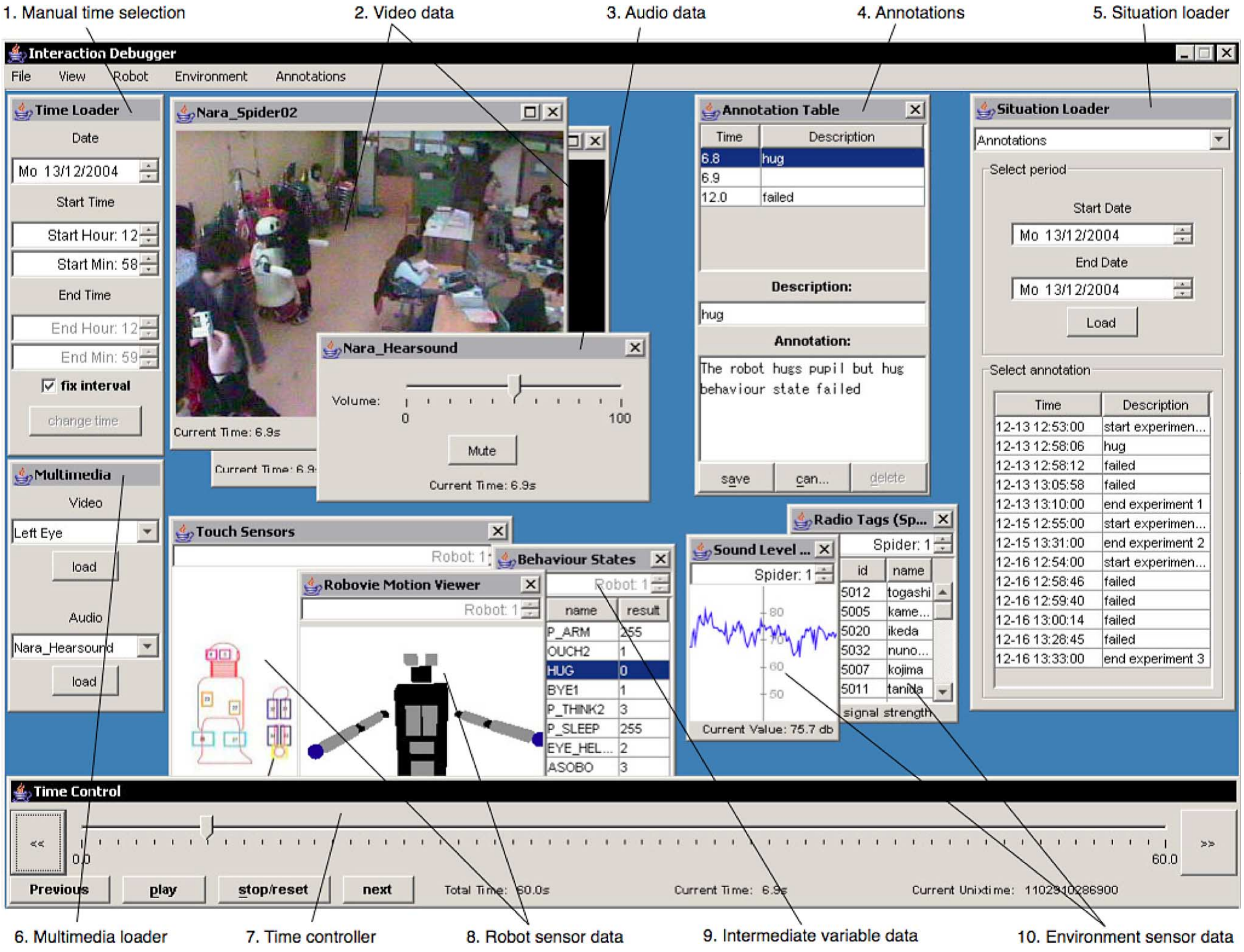
Results: -

Fig. 5. Filter-based search function.

B. End-User Functionalities

In this section, we elaborate how the Interaction Debugger's main functionalities are made available to end-users. We divide them into navigation, browsing, searching, analyzing, and annotating and describe each in detail. The functionalities apply to the controlled mode of the Interaction Debugger since the realtime mode only aims to show data.

1) Browsing: Whenever someone wants to analyze data, first he/she needs to know when data are available. The Interaction Debugger incorporates a browser (see Fig. 4) that gives an overview of the availability of all data types relative to time. The initial view provides an overview based on day precision, from



Date: Mo 13/12/2004

Start Time: Start Hour: 12, Start Min: 58

End Time: End Hour: 12, End Min: 59

☑ fix interval

change time

Current Time: 6.9s

Multimedia — Video: Left Eye (load); Audio: Nara_Hearsound (load)

Robot: 1

name	result
P_ARM	255
OUCH2	1
HUG	0
BYE1	1
P_THINK2	3
P_SLEEP	255
EYE_HEL...	2
ASOBO	3

Spider: 1

Current Value: 75.7 db

Spider: 1

id	name
5012	togashi
5005	kame...
5020	ikeda
5032	nuno...
5007	kojima
5011	tanida

Annotation Table

Time	Description
6.8	hug
6.9	
12.0	failed

Description: hug

Annotation: The robot hugs pupil but hug behaviour state failed

save | can... | delete

Situation Loader: Annotations

Select period — Start Date: Mo 13/12/2004; End Date: Mo 13/12/2004; Load

Select annotation

Time	Description
12-13 12:53:00	start experimen...
12-13 12:58:06	hug
12-13 12:58:12	failed
12-13 13:05:58	failed
12-13 13:10:00	end experiment 1
12-15 12:55:00	start experimen...
12-15 13:31:00	end experiment 2
12-16 12:54:00	start experimen...
12-16 12:58:46	failed
12-16 12:59:40	failed
12-16 13:00:14	failed
12-16 13:28:45	failed
12-16 13:33:00	end experiment 3

Total Time: 60.0s Current Time: 6.9s Current Unixtime: 1102910286900

Fig. 6. User Interface of Interaction Debugger.

where the user can zoom to specific days or even hours. Data types can be sorted either by modality or source. Furthermore, one can filter out certain modalities or sources to browse more purposefully.

2) Searching: During the analysis process, one is often interested in searching for situations in which useful/interesting data are available. In the HRI, for example, this could be a situation in which there are people around the robot or a robot performs certain behavior. The Interaction Debugger provides several data search methods: annotation search, behavior search, filter search, and query-based search.

1) Annotations can be regarded as bookmarks with descriptions made by a user when commenting on a certain event. The annotation search function gives an overview of all the bookmarks and makes it possible to show them in a specified time interval. Refer to Section III-B5 for more details on annotations.
2) The next search function is based on robot behaviors. A user can select a specific behavior and a time interval for which all the events in which the specified behavior occurs are shown. The behavior search function requires that the robot store active behavior states.
3) The most powerful search function in the Interaction Debugger is filter-based searching (see Fig. 5). When this search function is activated, every data frame obtains a bar at the bottom in which a user can select filters for the data presented in the frame. In this way, a user can combine filters for multiple data types to perform a search operation.
4) Professional users experienced with SQL scripting sometimes feel comfortable manually creating search queries. For them, we have implemented a search function that helps compose SQL queries for available data types.

3) Navigation: Navigating data in the controlled mode is based on the selection of a time interval, which can vary from 1 min to multiple days, and depends on end-user preferences and computer performance limitations. The user interface includes a time selection panel where one can manually choose a specific date as well as start and end times (Fig. 6.1). After selecting an interval, users can navigate through the data by a time controller that supports regular playback, frame-by-frame skipping, and mouse dragging. While controlling the time, an information window communicates the current interval and time.

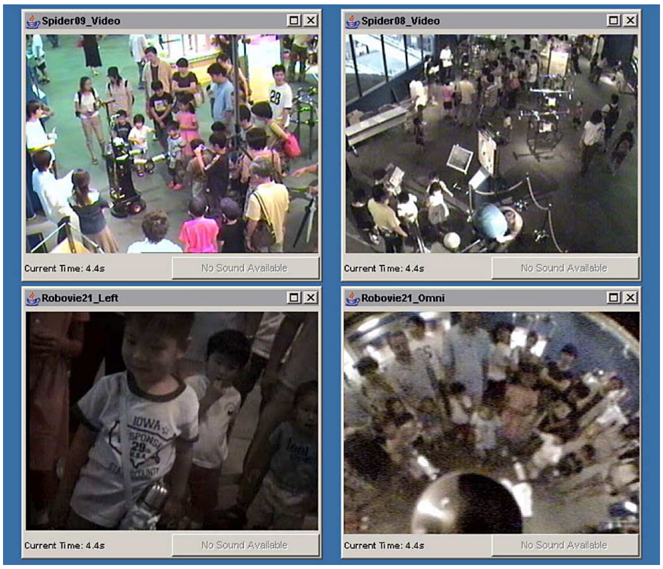


Fig. 7. Video data from multiple camera sources: Environment, robot eye, and omnidirectional cameras.

4) Analyzing: A core functionality of the Interaction Debugger is presenting data comprehensibly using visualizations tailored to the type of data. When using the software, one can open windows for every data type. All data types are accessible from the menu bar of the Interaction Debugger's main window, where they are sorted based on their modality. Fig. 6 shows a number of example windows that can be loaded:

1) *Figs 6.2 and 6.3:* Video and audio windows that can be simultaneously loaded from multiple sources (see Fig. 7).
2) *Fig. 6.8:* Visualizations of the robot's touch and motion sensors. The former blinks when the robot is touched. The latter are visualized by a 3-D model of the robot that resembles its motion. Other available windows for robot sensor data present data from ultrasonic distance sensors (see Fig. 8) and RFID tag readers.
3) *Fig. 6.9:* Active behavior states of the robot, which are one of its intermediate variables. The result of each behavior state is displayed, indicating success or failure.
4) *Fig. 6.10:* 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 presents data from distance sensors (see Fig. 11, Step 1).

Our implementation of the Interaction Debugger incorporates data presentation windows optimized for Robovie. To increase comprehensibility, we present the robot's sensor data on graphical representations of Robovie, as demonstrated in Fig. 8. The Interaction Debugger also features standard presentation styles including tables and line charts. For such textual data as behavior states and RFID tag readings, we use tables (Figs. 6.9 and 6.10). For single sensor values as sound levels, we use line charts.

5) Annotation: To aid the analysis of the HRIs, an annotation window (Fig. 6.4) has been incorporated in the Interaction Debugger. Inspired by existing audio and video annotation software [14]–[16], this feature allows users to describe every frame of the data collection. For example, researchers could use this functionality to make detailed descriptions of human behavior during empirical studies. Moreover, it can function as a useful bookmark and search method, as explained before.

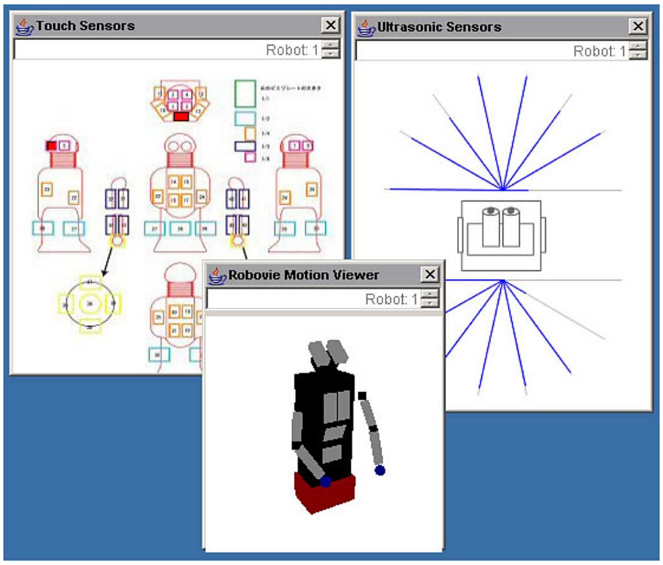


Fig. 8. Graphical representations of robot sensors.

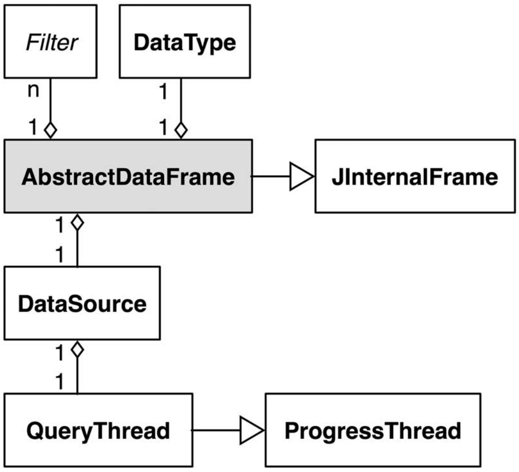


Fig. 9. UML diagram of extendible data frame.

C. Extendibility

Since the Interaction Debugger might be used for different types of robots in other situations, it has been designed with a modular software architecture. The graphical presentation of data and their underlying management have been clearly separated, making it easy for intermediate developers to modify or design new presentation styles. Moreover, it enables easy implementation of new data types by extending the AbstractDataFrame class. Fig. 9 shows a unified modeling language (UML) model that describes the structure of an

AbstractDataFrame object. It has a DataSource object that manages the retrieval of data from a database in the controlled mode or a TCP/IP socket in the realtime mode. It also contains a DataType object that, in turn, contains information about the data frame: name, modality, source name, and database table. Finally, the AbstractDataFrame object has a set of filter objects that are used for the filter-based search function.

IV. Case Studies

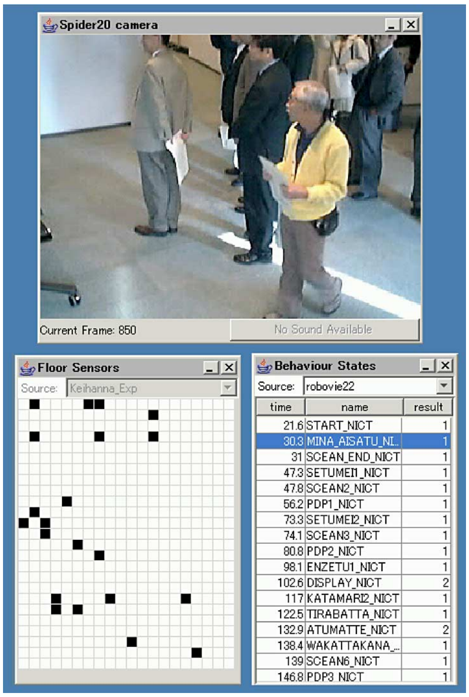
Integral analysis of the HRI can be used at several stages in the development process of interactive robots. To demonstrate how, we consider case studies related to the development of an interactive humanoid robot, Robovie, and its smaller variant, Robovie-M. For each case study, a step-by-step description will clarify how the Interaction Debugger was employed for data analysis.

A. Development of a Sensor Network

A technique for realizing intelligent behavior with a robot that is being used more and more is implementing a sensor network around a robot. In this way, the robot can extend its sensing capabilities beyond its own body and use sensors in the environment to analyze people. This case study is related to the development of a sensor network in an environment where Robovie had to interact with many people standing around it. The goal was to make communication more natural between Robovie and the people. One element was looking in the direction of people while talking with them. Moreover, estimating people's behavior would be helpful to make the robot interact with them accordingly.

We decided to use pressure sensors in the floor around Robovie to realize the localization and behavior estimation of people. After installing these sensors, the challenge was to develop an algorithm capable of analyzing sensor contacts created by the crowds and to discriminate people from this. The next step would then be to make it detect walking patterns and map such behaviors as "approaching," "conversing," "passing by," etc.

We developed this algorithm by collecting data from people who interacted with Robovie on a floor that contains pressure sensors and then using the integral analysis approach to find patterns. The types of data collected included video, floor sensor data, and behavior states of the robot. The chosen environment was an office building in which the robot interacted with groups of visitors. After the experiment, a developer analyzed the data using the Interaction Debugger to find patterns in the floor sensor data (see Fig. 10) by comparing the video in which he observed the people moving on the floor with the floor sensors. He also considered the behavior states of the robot to find correlations with human behavior, i.e., how they reacted to it. This could be helpful for a behavior estimation algorithm.



Current Frame: 850

No Sound Available

Source: Keihanna_Exp

Source: robovie22

time	name	result
21.6	START_NICT	1
30.3	MINA_AISATU_NI..	1
31	SCEAN_END_NICT	1
47.3	SETUMEI1_NICT	1
47.8	SCEAN2_NICT	1
56.2	PDP1_NICT	1
73.3	SETUMEI2_NICT	1
74.1	SCEAN3_NICT	1
80.8	PDP2_NICT	1
98.1	ENZETU1_NICT	1
102.6	DISPLAY_NICT	2
117	KATAMARI2_NICT	1
122.5	TIRABATTA_NICT	1
132.9	ATUMATTE_NICT	2
138.4	WAKATTAKANA_...	1
139	SCEAN6_NICT	1
146.8	PDP3 NICT	1

Fig. 10. Case study A.

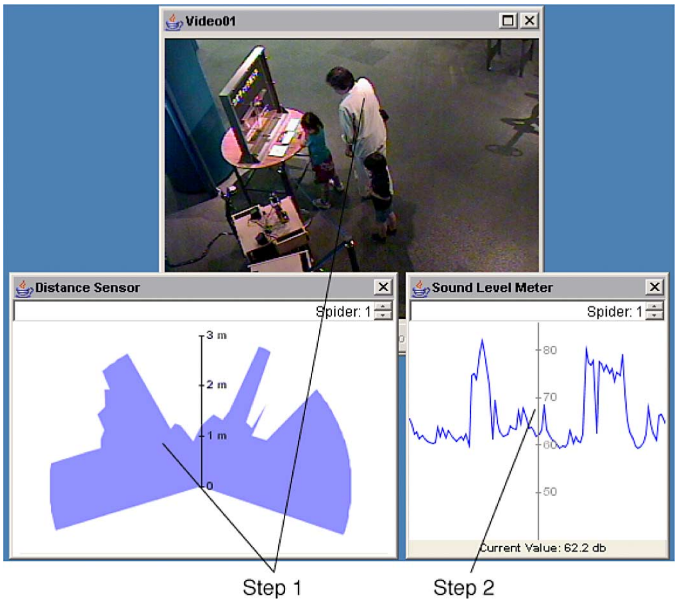
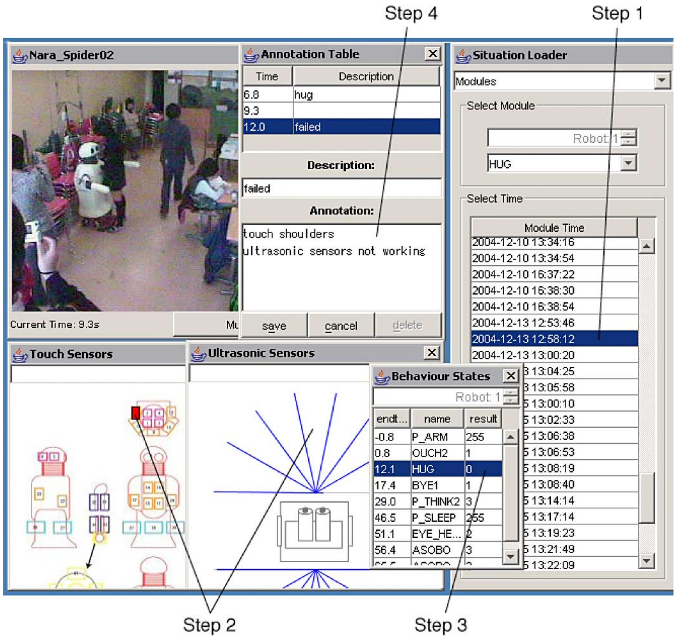


Fig. 11. Case study B.

B. 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 was part of our previous research activities [20]. Since Robovie-M was programmed to explain exhibits to visitors for this trial (see Fig. 11), 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



Current Time: 9.3s

Time	Description
6.8	hug
9.3	
12.0	failed

Description: failed

Annotation:

touch shoulders
ultrasonic sensors not working

Modules

Select Module: Robot: 1; HUG

Select Time

Module Time
2004-12-10 13:34:16
2004-12-10 13:34:54
2004-12-10 16:37:22
2004-12-10 16:38:30
2004-12-10 16:38:54
2004-12-13 12:53:46
2004-12-13 12:58:12
2004-12-13 13:00:20
3 13:04:25
3 13:05:58
5 13:00:10
5 13:02:33
5 13:06:38
5 13:06:53
5 13:08:19
5 13:08:40
5 13:14:14
5 13:17:14
5 13:19:23
5 13:21:49
5 13:22:09

Robot: 1

endt...	name	result
-0.8	P_ARM	255
0.8	OUCH2	1
12.1	HUG	0
17.4	BYE1	1
29.0	P_THINK2	3
46.5	P_SLEEP	255
51.1	EYE_HE...	2
56.4	ASOBO	3

Fig. 12. Case study C.

environmental situation is different, these thresholds have to be set manually.

The realtime mode of the Interaction Debugger was employed in this situation to optimize the presence detection thresholds. The robot developer used the following method (see Fig. 11):

1) *Step 1:* Using visualization of the infrared sensor, for 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.
2) *Step 2:* Sound level meter visualization was analyzed to set the voice detection threshold of the robot. In the picture, peaks represent human speech.
3) Using the Interaction Debugger, the developer successfully optimized the presence detection mechanism in a relatively short time span.

C. Evaluation of Hugging Behavior

During a field trial at a Japanese elementary school, Robovie was positioned in a classroom for 18 days 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 [10].

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

A robot developer used the Interaction Debugger to analyze the data recorded during 3 days of trials to debug the hugging behavior of the robot. His method can be summarized as follows (see Fig. 12):

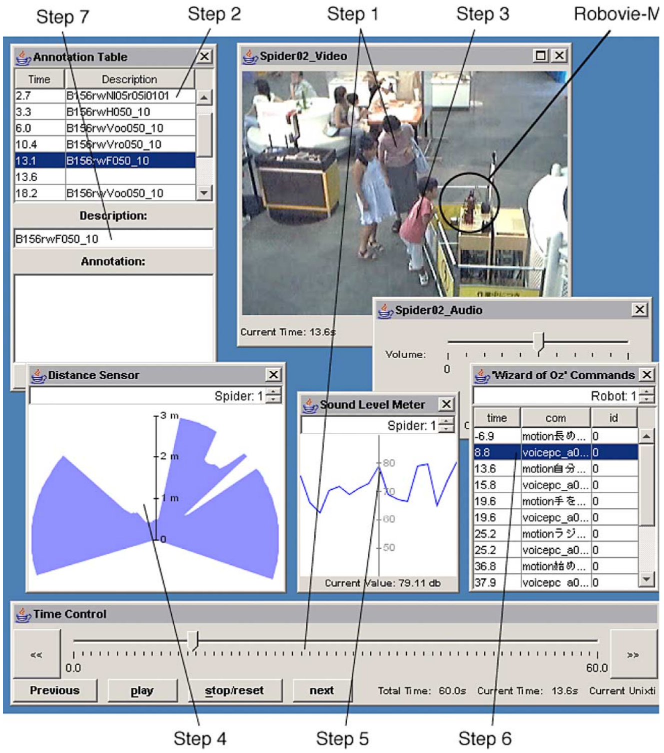
1) *Step 1:* In the behavior-based situation loader, the user selected hug behavior to retrieve a list of all hugging events. By double clicking on an entry, the corresponding time is automatically loaded. Data windows were loaded using the menu.
2) *Step 2:* For each event, he analyzed the robot's touch sensors and the distance sensor conditions 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 Interaction Debugger's time controller enabled him to study the data frame-by-frame.
3) *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.
4) *Step 4:* Finally, for each event, he annotated the success of the hug and the corresponding sensor conditions. Twenty percent of the hugs were not successful. The ultrasonic sensor window revealed the robot's failure 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 consider it effective analysis.

D. Studying Human Behavior in an Open Field

For the same field trial at the Osaka Science Museum as in Case Study A, a researcher with a background in psychology carried out an empirical study on the behavior of people who interacted with Robovie-M to learn how the crowd around a robot influences the way people react to it. This knowledge might eventually be useful 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 annotations using a code protocol. In this case, he considered every event where someone interacted 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 and 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.* [21]. Today, FACS is widely used for facial emotion recognition. Coding data by hand enables the analyst to use an exploratory approach and simultaneously get quantifiable results.

The factors that played an important role were the positions of people and the environmental sound level. Both provide information about crowdedness. He used the Interaction Debugger as a tool to analyze how these factors influenced human behavior toward the robot by the following method (see Fig. 13):



Time	Description
2.7	B156rwNl05r05i0101
3.3	B156rwH050_10
6.0	B156rwVoo050_10
10.4	B156rwVro050_10
13.1	B156rwF050_10
13.6	
18.2	B156rwVoo050_10

Description: B156rwF050_10

Annotation:

Current Time: 13.6s

Current Value: 79.11 db

time	com	id
-6.9	motion長め...	0
8.8	voicepc_a0...	0
13.6	motion自分...	0
15.8	voicepc_a0...	0
19.6	motion手を...	0
19.6	voicepc_a0...	0
25.2	motionラジ...	0
25.2	voicepc_a0...	0
36.8	motion始め...	0
37.9	voicepc a0...	0

Total Time: 60.0s Current Time: 13.6s Current Unixti

Fig. 13. Case study D.

1) *Step 1:* First he found events where people approached the robot by rapidly skipping through the scenes by moving the timeline slider. Whenever noticing people approaching the robot, he used the play function to scrutinize the interaction event.
2) *Step 2:* For each interaction event, he coded its start and end times, the person's ID (e.g., B156), adult/child, alone/group, and gender in the annotation window.
3) *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, imitation, waving, bending, touching, etc.). To consider the human behavior in detail, he viewed the data frame-by-frame using the time controller.
4) *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 humans and the robot.
5) *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 of crowdedness.
6) *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.
7) *Step 7:* For each interaction event, he recorded every human action and the results from Steps 2–4 as coded annotation.
8) *Step 8:* After using the 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 could be used later so that the robot can automatically infer that people are interested in it by measuring crowdedness and their movements. The results of this study were published by Nabe *et al.* [22].

V. Discussion

A. Contributions

This paper presented a multimodal approach to analyze the HRI, 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 [23]. In this research, interhuman interaction was analyzed in the presence of a humanoid robot.

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

B. Evaluation of the Approach

Unlike the evaluation of a method, evaluating a new methodology is difficult. 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 into the following methods: "showing sensory information" and the "integration of multimodal information." To evaluate the effectiveness of the first method, for instance, we could conduct a controlled experiment that compares analysis results with/without 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 multimodal information, we compared the use of the Interaction Debugger with a common audio/video controller 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 does not lead to a clear impression about the validity of the interaction debugging approach as a whole. Therefore, instead of conducting such experiments, in this paper we focused on the introduction of our integral approach by presenting case studies.

C. Evaluation of Software Usability

Since the 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.* [24], "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 the 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" [25]. Furthermore, the case studies were part of a user-centered method to optimize the Interaction Debugger's user interface [26]. 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 the 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 graphical user interfaces (GUIs), new users only needed a brief explanation of the different functions to get started. For nonnovice 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 design problem worth mentioning concerns the organization of windows in the Interaction Debugger. Because the number 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.

D. Evaluation of Software Performance

The Java language used for developing the Interaction Debugger software is sometimes considered to lack efficiency for high-performance applications. We managed to overcome this limitation by using several techniques such as preprocessing of data and proper reusing of objects to prevent excessive garbage collection.

When loading a time interval as explained in Section III-B3, the Interaction Debugger preprocesses the data to optimize navigation performance. We use an efficient dichotomy search algorithm to quickly locate the correct data when moving the playhead. As an example, when we open two videos, one audio, one ultrasonic sensor, one behavior-module, and one motion panel, for a 5-min data duration, memory consumption is approximately 35 MB. Everything worked in realtime when we run this configuration on a computer with 512-MB memory with an ordinary Pentium IV CPU.

E. Limitations

In this paper, we only demonstrated four 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, the applicability and effectiveness for large-scale development remains unclear.

At the moment, the generalizability of our approach is also still unknown. Since it was only tested in a limited number of applications, we cannot determine in which cases the approach will be applicable and effective and in which cases it will not be. We believe that using an interaction debugging approach for more widespread robot development activities will foster a better view on this. We would, thus, 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 modalities required for 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 these data.

VI. Conclusion

We believe that an integral approach to analyze the HRI, which involves multiple modalities, has been inevitable since the complexity of the interaction between robots and humans in everyday situations has increased dramatically. We developed the Interaction Debugger software to aid the interaction data analysis. This tool allows us to conduct interdisciplinary projects that investigate the HRI by offering an environment that encourages collaboration between robot developers and psychologists. We demonstrated the practical applicability of the interaction debugging approach by addressing four different case studies that demonstrated the necessity of observing different modalities. In all of them, using the Interaction Debugger led to an effective analysis of the HRI. However, we are aware that this only illustrates a limited number of applications and does not compare its effectiveness to other methodologies. The Interaction Debugger's functionality exceeds any of the conventional video data analysis tools and, hence, a comparison could not provide valid insights. Instead, we hope to inspire researchers to adopt comparable methods to move toward a more quantitative and, hence, scientific approach to the HRI.

Acknowledgment

The authors thank S. Nabe, Y. Koide, and J. Lin for their valuable contributions toward the development and optimization of the Interaction Debugger.

REFERENCES

[1] R. Sakagami, C. Watanabe, S. Aoyama, N. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 2478–2483.

[2] M. Fujita, "AIBO: Toward the era of digital creatures," *Int. J. Robot. Res.*, vol. 20, no. 10, pp. 781–794, 2001.

[3] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, "Robovie: An interactive humanoid robot," *Int. J. Ind. Robot*, vol. 28, no. 6, pp. 498–503, 2001.

[4] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami, and T. Siméon, "How may I serve you?: A robot companion approaching a seated person in a helping context," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 172–179.

[5] R. Gockley, J. Forlizzi, and R. Simmons, "Interactions with a moody robot," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 186–193.

[6] C. Breazeal, "Social interaction in HRI: The robot view," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 181–186, May 2004.

[7] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human–robot interaction," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 33–40.

[8] H. Ishiguro, T. Ono, M. Imai, and T. Kanda, "Development of an interactive humanoid robot 'Robovie' —An interdisciplinary approach," in *Robotics Research*, R. A. Jarvis and A. Zelinsky, Eds. Berlin, Germany: Springer, 2003, pp. 179–191.

[9] K. Dautenhahn and I. Werry, "A quantitative technique for analysing robot–human interactions," in *Proc. 2002 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 1132–1138.

[10] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro, "Interactive robots as social partners and peer tutors for children: A field trial," *J. Hum. Comput. Interact.*, vol. 19, no. 1–2, pp. 61–84, 2004.

[11] R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis, "Robox at Expo.02: A large-scale installation of personal robots," *Robot. Auton. Syst.*, vol. 42, no. 3–4, pp. 203–222, 2003.

[12] F. Tanaka, J. R. Movellan, B. Fortenberry, and K. Aisaka, "Daily HRI evaluation at a classroom environment: Reports from dance interaction experiments," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 3–9.

[13] J. Burke, R. Murphy, D. Riddle, and T. Fincannon, "Task performance metrics in human–robot interaction: Taking a systems approach," presented at Perform. Metrics Intell. Syst. Workshop, Gaithersburg, MD, 2004.

[14] L. P. Noldus, R. J. Trienes, A. H. Hendriksen, H. Jansen, and R. G. Jansen, "The Observer Video-Pro: New software for the collection, management, and presentation of time-structured data from videotapes and digital media files," *Behav. Res. Methods, Instrum. Comput.*, vol. 32, pp. 197–206, 2000.

[15] F. Quek, Y. Shi, C. Kirbas, and S. Wu, "VisSTA: A tool for analyzing multimodal discourse data," presented at the 7th Int. Conf. Spoken Lang. Process., Denver, CO, 2002.

[16] M. Kipp and A. Anvil, "Generic annotation tool for multimodal dialogue," in *Proc. 7th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, 2001, pp. 1367–1370.

[17] T. Kanda, H. Ishiguro, M. Imai, and T. Ono, "Development and evaluation of interactive humanoid robots," *Proc. IEEE*, vol. 92, no. 11, pp. 1839–1850, Nov. 2004.

[18] Y. Sumi, S. Ito, T. Matsuguchi, S. Fels, and K. Mase, "Collaborative capturing and interpretation of interactions," in *Proc. Pervasive 2004 Workshop Mem. Shar. Exp.*, pp. 1–7.

[19] T. Kooijmans, T. Kanda, C. Bartneck, H. Ishiguro, and N. Hagita, "Interaction debugging: An integral approach to analyze human–robot interaction," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 64–71.

[20] T. Nomura, T. Tasaki, T. Kanda, M. Shiomi, H. Ishiguro, and N. Hagita, "Questionnaire-based research on opinions of visitors for communication robots at an exhibition in Japan," presented at the Int. Conf. Hum.–Comput. Interact., 2005.

[21] P. Ekman and W. V. Friensen, *Manual for the Facial Action Coding System and Action Unit Photographs*. Palo Alto, CA: Consulting Psychologists, 1978.

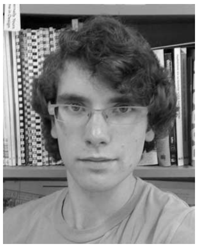
[22] S. Nabe, T. Kanda, K. Hiraki, H. Ishiguro, K. Kogure, and N. Hagita, "Analysis of human behavior in an open field to improve communication robots," in *Proc. 1st ACM SIGCHI/SIGART Conf. Hum.–Robot Interact.*, 2006, pp. 234–241.

[23] S. Nabe, T. Kanda, K. Hiraki, H. Ishiguro, and N. Hagita, "Human friendship estimation model for communication robots," presented at the 2005 IEEE-RAS Int. Conf. Humanoid Robots, 2005.

[24] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human–Computer Interaction*. New York: Wiley, 2002, ch. 1.

[25] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human–Computer Interaction*. Reading, MA: Addison-Wesley Longman, 1998.

[26] D. Norman and S. Draper, *User Centered System Design; New Perspectives on Human–Computer Interaction*. Mahwah, NJ: Erlbaum, 1986.



Tijn Kooijmans received the Bachelor of Science degree in industrial design from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2006.

In 2005, he was a Research Intern at the Intelligent Robotics and Communication Laboratories, Advanced Telecommunications Research Institute International, Kyoto, Japan. He, recently, cofounded Studio Sophisti, Amsterdam, The Netherlands, where he is engaged in research on interactive product design. His current research interests include entertainment computing and affective communication in human-computer interaction.

Takayuki Kanda (M'03) received the B.Eng., M.Eng., and Ph.D. degrees in computer science from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2003, respectively.

From 2000 to 2003, he was an Intern Researcher at the Advanced Telecommunications Research Institute International (ATR), Media Information Science Laboratories, Kyoto. He is currently a Senior Researcher at the Intelligent Robotics and Communication Laboratories, ATR, Kyoto. His current research interests include intelligent robotics, human–robot interaction, and vision-based mobile robots.

Dr. Kanda is a member of the Association for Computing Machinery, the Robotics Society of Japan, the Information Processing Society of Japan, and the Japanese Society for Artificial Intelligence.

Christoph Bartneck received the Diplom Industrial-Design Informatik, in 1997, the Master of Technological Design and the Ph.D. degrees in user–System interaction, in 2000 and 2002, respectively.

He is currently an Assistant Professor in the Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands. Prior to this, he had been with several companies including the Technology Center of Hannover, Germany, LEGO, Denmark, Eagle River Interactive, USA, Philips Research, The Netherlands, and Advanced Telecommunications Research Institute (ATR), Japan.

He is the author or coauthor of several papers published in international journals. His current research interests include human–robot interactions, design science, and multimedia applications.

Hiroshi Ishiguro received the D.Eng. degree from Osaka University, Osaka, Japan, in 1991.

In 1991, he joined the Department of Electrical Engineering and Computer Science, Yamanashi University, Yamanashi, Japan, as a Research Assistant. In 1992, he joined the Department of Systems Engineering, Osaka University, Osaka, Japan, as a Research Assistant. In 1994, he was an Associate Professor in the Department of Information Science, Kyoto University, Kyoto, Japan, where he was engaged in research on distributed vision using omnidirectional cameras. From 1998 to 1999, he was with the Department of Electrical and Computer Engineering, University of California, San Diego, as a Visiting Scholar. Since 1999, he has been a Visiting Researcher at the Advanced Telecommunications Research Institute International (ATR), Media Information Science Laboratories, Kyoto, Japan, where he has been engaged in research on the interactive humanoid robot, Robovie. In 2000, he joined the Department of Computer and Communication Sciences, Wakayama University, Wakayama, Japan, as an Associate Professor and became a Professor in 2001. He is currently a Professor in the Department of Adaptive Machine Systems, Osaka University, Osaka, Japan, and a Group Leader at the Intelligent Robotics and Communication Laboratories, ATR, Kyoto, Japan. His current research interests include omnidirectional vision, distributed vision, sensor networks, humanoid robots, and android robots. In the last several years, he has mainly focused on the development of humanoids and androids. The most successful robot he has made was exhibited in Nagoya World Expo 2005, for which he is considered the father of the world's first android. In addition, he has developed the geminoid that is a tele-operated android. His activities have been introduced by almost all major TVs and newspapers in the world.

Recently, he has been selected as one of the eight scientists who change our life by CMM.com.

Norihiro Hagita (M'85–SM'99) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Keio University, Tsuruoka City, Japan, in 1976, 1978, and 1986, respectively.

During 1978–2001, he was with the Nippon Telegraph and Telephone Corporation (NTT). He, then, joined the Advanced Telecommunications Research Institute International (ATR) to establish the ATR Media Information Science Laboratories and the ATR Intelligent Robotics and Communication Laboratories, in 2001 and 2002, respectively. His current research interests include communication robots, network robot system, interaction media, and pattern recognition.

Dr. Hagita is a Fellow of the Institute of Electronics, Information, and Communication Engineers, Japan. He is a member of the Robotics Society of Japan, the Information Processing Society of Japan, and The Japanese Society for Artificial Intelligence. He is a Co-Chair for the IEEE Technical Committee on Networked Robots.