Grace and George: Social Robots at AAAI

Rachel Gockley, Reid Simmons, Jue Wang, Didac Busquets, Carl DiSalvo
Carnegie Mellon University

Kevin Caffrey, Stephanie Rosenthal, Jessica Mink, Scott Thomas, William Adams
Thomas Lauducci, **Magda Bugajska**, **Dennis Perzanowski**, **Alan Schultz**
Naval Research Laboratory

Abstract

Grace and George have been past entrants in the AAAI Robot Challenge. This year, however, we chose to integrate our more recent work in the field of human-robot interaction, and designed a system to enter the "Open Interaction" category. Our goal was to have two robots at the AAAI National Conference, with one acting as an "information kiosk," telling about the Conference and giving directions, and the other acting as a mobile escort for people. This paper discusses the system we envisioned, as well as what we were able to achieve at the Conference.

Introduction

New to the AAAI National Conference this year was the "Open Interaction" event. The goal of this event was simply to entertain people, by having a robot engage in free-form interaction with humans. The only requirement was that the robot have some form of AI at its core.

Research teams at Carnegie Mellon University (CMU) and the Naval Research Laboratory (NRL) decided to address the Open Interaction task by combining past efforts on the AAAI Robot Challenge with each group's recent research efforts.

Prior work

Grace and George represent the combined effort of researchers at Carnegie Mellon University and at the Naval Research Laboratory, as well as past effort by other groups. In particular, Grace and George emerged from past effort on the AAAI Robot Challenge, human-robot interaction research on Valerie the Roboceptionist, and language processing work with NAUTILUS.

AAAI Robot Challenge

At the past two AAAI National Conferences, the research team responsible for Grace and George has competed in the AAAI Robot Challenge (Simmons *et al.* 2003b; 2003a). The task of the Challenge is for a robot to attend the Conference as a participant—register at the registration booth, navigate to a specified location, and give a technical talk on itself.

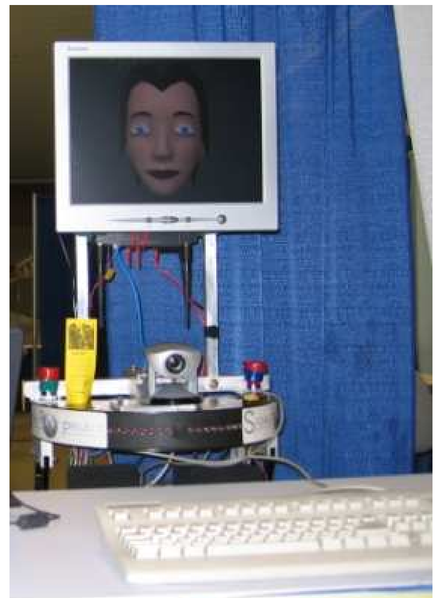

Figure 1: Grace behind her desk.

Successful completion of the Challenge requires the integration of many technologies, including localization and navigation in a dynamic environment, speech and gesture recognition, natural language understanding, and social interaction with people. All of these technologies were addressed at least in part with Grace and George at various Challenge events. In particular, the research teams worked to develop an architecture and infrastructure that would enable the numerous pieces of software to be integrated into a complete system. That basic infrastructure was used in this year's entry in the Open Interaction event.

Valerie the Roboceptionist

Valerie the Roboceptionist (short for "robot receptionist") was introduced to the Carnegie Mellon University community in November, 2003. A permanent installation in the entranceway to one of the University's buildings, Valerie combines useful functionality—giving directions, looking up weather forecasts, etc.—with an interesting and compelling character. Her personality and character were developed through a collaboration with the Drama Department at Carnegie Mellon. She tells stories of her life, which evolve over time, to spark interest in her visitors. Similar to George (Fig. 2), Valerie is a B21r mobile robot with a moving flat-panel monitor mounted on top, which displays a graphical face. By panning the display, the robot can "look" as much as 120 degrees to either side. Though the base is mobile, Valerie remains stationary inside a custom-built booth.

Valerie uses automatic text-to-speech generation to talk to visitors. Visitors, however, must type on a keyboard in order to communicate with Valerie. Though speech would be more natural, keyboard input is easier to control and is more reliable than speech-to-text systems. Also, due to Valerie's placement in a busy hallway, speech recognition would be severely degraded due to large amounts of background noise.

Aside from telling stories of her life, Valerie can look up office numbers and give directions to campus buildings. Office numbers are determined through function calls on the internal network. Directions to the various buildings were manually generated, and are simply retrieved as required.

The language processing system that Valerie uses is a simplistic rule-based pattern-matching "chatbot," modified from Aine (www.neodave.civ.pl/aine). Unlike a robust parser, Aine is completely ignorant of the way language works, and it will respond to matched words and phrases without regard for the contexts in which they are found. On the other hand, it can handle ungrammatical sentences, sentence fragments, and even many misspellings, if appropriate rules have been added to the Aine database.

NAUTILUS

In contrast to Aine, NAUTILUS (Wauchope 1994), the natural language understanding system developed at NRL, produces a so-called "deep parse" from its (text string) inputs, utilizing syntactic and semantic information from its lexicons. Semantic information is used initially to reduce the number of possible parses of the input string (Wauchope 1990). An additional module resolves anaphora and quantified noun phrases. Towards the end of processing, a quantified logic expression is constructed (in Lisp) that encodes, in a general form, the system's understanding of the input string. Execution of the expression results in calls to subroutines that correspond to the main concepts named in the input string, usually by verb of the sentence: move, turn, tell, stop, continue, etc. These subroutines in turn must be linked to code supplying the appropriate application-dependent functionality.

Unfortunately, the performance of NAUTILUS can degrade abruptly in the face of unknown words and sentence types, though a reasonable guess can be made for the category of certain word types, such as nouns. Furthermore, adding new grammar rules, and even lexical entries, when they are of a novel type, can be a time-consuming process. Unlike Aine, NAUTILUS is designed for results that are precise but of narrow coverage; Aine sacrifices precision for (relative) ease of coverage.

Figure 2: George with mounted keyboard and printer.

Goals

To address the Open Interaction Event, we decided to build on our previous research and design two robots, one stationary and one mobile.

Information Kiosk

Drawing heavily on the work done for Valerie, we decided to have one robot, Grace, act as an information kiosk. The robot would be stationary behind a desk. People would be able to approach the robot and interact with it by typing on a keyboard on the desk. The robot would be able to answer queries about the Conference and the surrounding area, give directions within the conference center, and also engage in idle chat.

Escorting

In addition to the information kiosk robot, we decided to have a second robot, George, which would have the same capabilities as the kiosk robot but also be mobile. George would have the added capability to escort people to their desired location. When within range of each other, the two robots would communicate, such that Grace could summon George to escort someone from Grace's desk, provided George was not currently preoccupied.

Hardware

Grace (Fig. 1) is built on an iRobot B21 mobile robot, while George (Fig. 2) is built on an iRobot B21r. Each robot displays an expressive computer-animated face on a flat-panel LCD monitor. Because George was expected to interact with people while in free space and while escorting, George's monitor was mounted on a pan unit, while Grace's position behind a desk allowed a fixed mount. Both robots have built-in touch, infrared, and sonar sensor arrays. Near the base of each is a SICK scanning laser range finder, which provides a

Figure 3: User input is displayed on the robot's "head," directly beneath the face.

180-degree field of view. Both robots have cameras on board for color tracking, though this capability was not used in the Open Interaction task.

As with Valerie, both Grace and George required keyboards for user input. Since Grace was to remain behind a desk, a standard computer keyboard was simply placed on the desk. To allow George to be mobile, however, a small keyboard was mounted to the mezzanine of the robot, a bit below the monitor (Fig. 2). The positioning of the keyboard attempted to take into account both comfortable typing height for most people, as well as a comfortable conversational space.

Both robots were also equipped with small-format thermal receipt printers. These were intended to be used to print copies of directions for visitors. For Grace, the printer was just placed on the table. George's printer was mounted beneath his keyboard, and oriented such that the output from the printer was redirected above the keyboard, where the visitor could easily take it.

Software Architecture

The basic software architecture follows the same design that we used in previous years for our entries in the Robot Challenge. Each robot runs a set of independent processes that communicate via IPC message passing (http://www.cs.cmu.edu/∼IPC). Processes could be independently started and stopped through the Microraptor system (http://gs295.sp.cs.cmu.edu/brennan/mraptor/).

Much of the software used in Grace and George was developed previously, either for previous AAAI Challenge events, or for Valerie. New this year were a modified interface, a "dialogue manager" to handle user input, an improved direction-giver, and escorting capabilities.

Interface

For interaction with humans, both robots display an expressive, animated face on their LCD "heads." The faces are used for both emotional expression and for simple gestures (head pointing and directing gaze), since the hardware lacks any conventional manipulators. Each robot uses the Festival (http://www.cstr.ed.ac.uk/projects/festival) text-to-speech system to automatically generate utterances. The

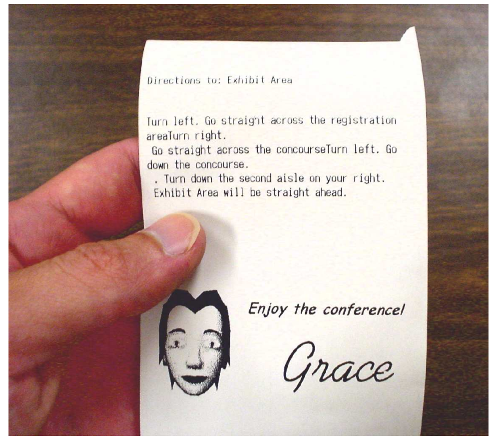

Figure 4: Printed directions from Grace.

process that controls the facial musculature ("expression") also performs automatic lip-syncing with the utterances. In addition, the utterances are displayed in cartoon-styled text balloons, in order to aid human understanding of the synthetic speech.

Previous appearances of Grace and George at AAAI Challenge events featured a speech interface. However, the speech-to-text software performed poorly both in noisy environments, and for speakers who had not trained a model nor were familiar with the bounds of the grammar. As a result, we decided to remove the speech-to-text component this year. Replacing this component with a conventional keyboard allowed Conference attendees to interact with the robot with much less confusion and misunderstood speech. To enable visitors to the robots to see what they were typing without the use of a second monitor (as with Valerie), user input was displayed, as it was typed, below the face on the LCD monitor (Fig. 3).

In addition to receiving spoken directions from the robots, visitors could request directions in printed form. Directions were then printed on the robot's thermal printer for the user to take along (Fig. 4). To reduce paper waste, directions were only printed for those visitors who explicitly asked for a printed copy.

Dialogue Manager

The dialogue manager is a finite state machine, which serves to dispatch user input to appropriate processes, perform database lookups for certain information, and manage a small amount of dialogue. Fig. 5 shows the basic architecture. User input is passed to the dialogue manager, which in turn passes it to the language processing module, which was built using a combination of NAUTILUS and Aine. The response from the language processing module is then returned to the dialogue manager. In some cases, the dialogue manager passes the response directly to the user. Other times, the response is a query for the dialogue manager to handle. Depending on the query, the dialogue manager will either generate a response via a database lookup (for Conference talks and local information), or will dispatch the query

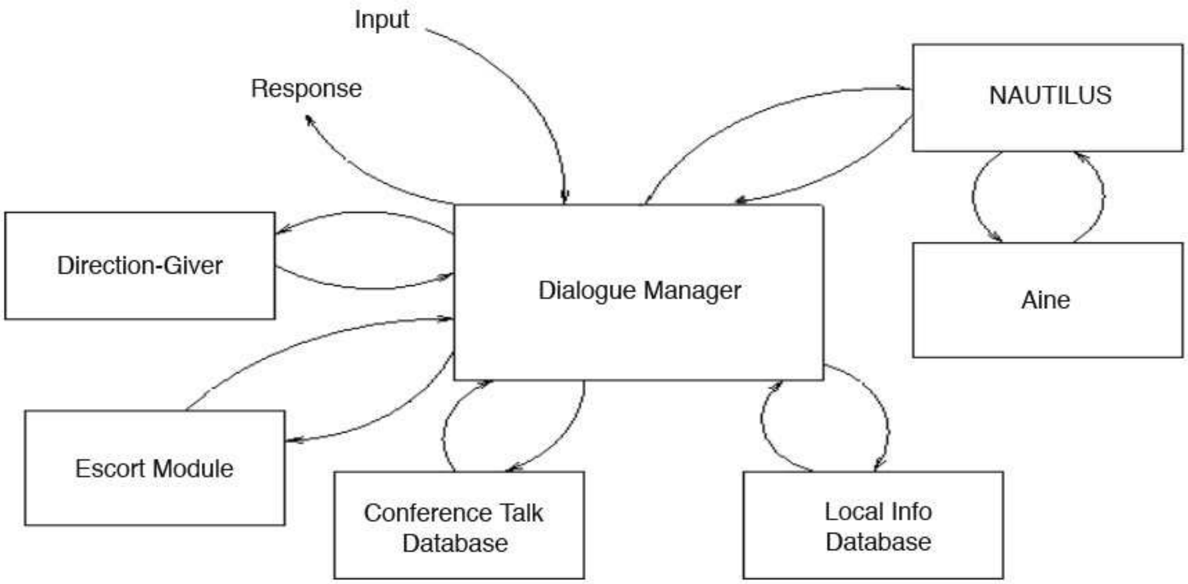

Figure 5: Dialogue Manager architecture

to the direction-giver or the escort module (discussed below).

Our original intent was to use Aine only as a fall-back module, for inputs that fell outside of the NAUTILUS grammar. However, there may also be sentences that NAUTILUS can parse and interpret, but still not have a good response for, because of potential gaps in the application-specific code that defines those responses. Furthermore, it is much easier to add venue-appropriate pattern/response pairs to Aine, and so for some classes of inputs we decided to ignore NAUTILUS's (valid) response, anticipating that it would be more generic than Aine's. On the other hand, we also had to make sure that NAUTILUS's response would be used, however generic, in those cases where Aine gave only an uninformative default response. During the conference runs, though, almost all of the input was handled by Aine rather than NAUTILUS.

Direction Giving

A metric map of the Conference area was generated in advance with Carmen (http://www-2.cs.cmu.edu/∼carmen/), and used as the basis for a handmade topological map. The direction giver uses the topological map to find a path and generate directions from the starting location to the destination. The map is laid out in a graph with vertices representing destinations in the map such as rooms, hallways, common areas, and points of interest. The edges represent lengths of paths that connect each destination on the map. In addition, the graph incorporates distance between points and angles between connected points to assist in direction generation.

In an enclosed environment, the direction giver can easily generate directions by referencing intersections of hallways and different hallways to turn down. In an outdoor environment, there are not as many objects or paths that are easily referenced. Therefore, points that represent landmarks, such as a flag pole or a clock tower, are given special landmark weight values. This allows the direction giver to give accurate directions in ambiguous environments by guiding the user according to the landmarks.

The direction generation algorithm works as follows. A path is found through the map by using Dijkstra's shortest-path algorithm. An initial pass is made down the entire path, splitting it into segments of straight paths. The assumption here is that humans give directions by telling each other to go down a certain segment of the path in a certain way, and then how to transition to the next segment of the path. Basic data, such as which way to turn, whether segments of the path go through ambiguous environments, and what landmarks to use to guide the user, are also collected during the pass. Directions for the different segments of the paths and directions for transitioning from one segment to another are generated on the fly during the first pass. Finally, the directions for the various segments are combined, and redundancies are cut out.

Translation of directions into spoken English relies on the distances and relative angles between points, as stored in the augmented topological map. When traversing the nodes to generate spoken directions, direction changes such as "turn left" or "make a slight turn right" were output based on the angle.

Escorting

The goal of the escort module was to respond to escort requests by determining whether the robot was currently free to escort someone, and if so, leading that person to his desired destination. This would require negotiating with the person, possibly moving to meet the person at a known location, tracking the person while navigating, and modifying the robot's speed as necessary so as not to lose the person.

The navigational tasks required for escorting were addressed in previous AAAI Challenge events with Grace and George. These tasks include map generation, dynamic localization, path planning, and navigation, and all were reused for this event with little modification.

For person tracking, we decided to simplify the task to tracking a colored hat that we would provide to the escortee. We developed a tracker based on the Swarthmore Vision Module (SVM) (Maxwell *et al.* 2002). The person tracker was designed to set the robot's maximum velocity based on how far away the person was—if the escortee lagged behind, then the robot would slow down to accommodate him. Unfortunately, due to time constraints, the people-tracker was not fully integrated into George (see below).

Visitors would be able to ask the escort robot directly to be lead to a destination. In addition, we wanted that visitors could ask for an escort from the information kiosk. To handle escort requests in this way, the kiosk robot would attempt to contact the escort through the wireless network. If the escort was out of wireless range, the kiosk could immediately report that the escort was unavailable. Otherwise, the escort would compute approximately how long it would need to return to a common meeting point (including time to finish a current escort task, if applicable), and send that time as a response to the kiosk. The kiosk could then ask the requesting person whether he wanted to wait that long, and send the response back to the escort.

Performance at AAAI

Grace was stationed behind a desk labeled "Information" for a significant portion of the Conference (about two full days). Much of her interactions with the public was comprised of 'small talk' about her origin and capabilities. She could also engage in conversations like the following, providing information on local restaurants, and how to find them:

> where is mexican food
speak("There is a restaurant called Chacho's about half a mile away.") speak("Would you like the address?")

> yes
speak("The address for Chacho's is") speak("18 S Almaden Ave San Jose, CA 95113")

> print that please
speak("Please take the directions")

> next one
speak("There is a restaurant called Pedro's Restaurant & Cantina a few miles away.") speak("Would you like the address?")

> no
speak("Ok")

In another example of interaction, Grace consults the Conference talk database and map to guide an attendee to the requested talk:

> what is your name
speak("I am Grace.")

> what is your last name?
speak("I don't have a last name. Computers have more efficient ways of addressing each other.")

> what is a good chinese restaurant
speak("There is a restaurant called Pagoda Restaurant about half a mile away.") speak("Would you like the address?")

> yes
speak("The address for Pagoda Restaurant is") speak("170 South Market St San Jose, CA 95113")

> where is manuela's talk
speak("Manuela Veloso has a talk in room A8.") speak("Would you like directions?")

> yes
speak("Turn left. Go straight across the registration areaTurn right.") speak("Turn left. Go straight down the concourse.") speak("A8 will be the first meeting room on the left.")

(Unfortunately, the last response also shows a text generation bug: an errant "Turn right" appears in the string.) All but one of the input/output pairs were handled by Aine rather than NAUTILUS.

The team's official "demo" time was near the end of the Conference. Because escorting was still not fully integrated, we stationed George outside of the exhibition hall, running the same "information kiosk" version of the software that Grace used. Some people even noticed that George and Grace were responding similarly to similar questions, and were disappointed not to see distinct behaviors. Nonetheless, George attracted a significant number of people throughout the day (Fig. 6), just as Grace did, at least at the start of the Conference. Because the researchers were still integrating components of the system as the Conference was starting, testing was ongoing, and Grace did show multiple periods of unresponsiveness, due to bugs in then Dialogue Manager and the "expression" process. Even then, we often noted a large number of people attempting to interact with her. Fewer people were approaching her toward the end of the Conference, though, possibly as a result of her earlier unresponsiveness. People often approached in groups, rather than one at a time.

George attempted one escort run during the Conference, at the request of a person interacting directly with George. George began escorting that person to the Exhibit Hall, but was unable to complete due to a steady stream of people blocking the door into the Hall. Also, because the people-tracking algorithm was never fully integrated, when George was escorting someone he was not watching where that per-

Figure 6: George interacting with a group of people.

son was nor trying to maintain a proper lead.

Discussion

Clearly, the system we envisioned suffered greatly due to a time crunch, resulting in reduced functionality from what we had planned (with escorting in particular) and in reduced reliability in the parts of the system that were put into place. However, we did learn several important things from the experience.

One thing we learned is that multi-institution software integration takes time. This is fairly obvious and well-known, but yet it was our largest problem this year (as with previous years).

Also, robotic kiosk systems must be extremely robust; otherwise, the public will tend to become annoyed and/or disinterested. We believe we saw evidence of this in the (apparent) reduced interest in Grace later in the Conference, following the periodic unresponsiveness she exhibited earlier.

Finally, we note that we ended up using NAUTILUS much less than we had expected. At first, the idea of using Aine purely as a fall-back system seemed very natural, knowing that it has a low precision rate. But we failed to take into consideration how much faster it would be to add venue-specific behaviors to Aine, rather than NAUTILUS.

References

Gockley, R.; Bruce, A.; Forlizzi, J.; Michalowski, M.; Mundell, A.; Rosenthal, S.; Sellner, B.; Simmons, R.; Snipes, K.; Schultz, A.; and Wang, J. 2004. Designing robots for long-term social interaction. Submitted to ICRA2005.

Maxwell, B. A.; Fairfield, N.; Johnson, N.; Malla, P.; Dickson, P.; Kim, S.; Wojtkowski, S.; and Stepleton, T. 2002. A real-time vision module for interactive perceptual agents. *Machine Vision and Applications*.

Simmons, R.; Bruce, A.; Goldberg, D.; Goode, A.; Montemerlo, M.; Roy, N.; Sellner, B.; Urmson, C.; Schultz, A.; Adams, W.; Bugajska, M.; MacMahon, M.; Mink, J.; Perzanowski, D.; Rosenthal, S.; Thomas, S.; Horswill, I.; Zubek, R.; Kortenkamp, D.; Wolfe, B.; Milam, T.; and Maxwell, B. 2003a. Grace and George: Autonomous robots for the AAAI robot challenge. Workshop paper for AAAI 2003.

Simmons, R.; Goldberg, D.; Goode, A.; Montemerlo, M.; Roy, N.; Sellner, B.; Urmson, C.; Schultz, A.; Abramson, M.; Adams, W.; Atrash, A.; Bugajska, M.; Coblenz, M.; MacMahon, M.; Perzanowski, D.; Horswill, I.; Zubek, R.; Kortenkamp, D.; Wolfe, B.; Milam, T.; and Maxwell, B. 2003b. Grace: An autonomous robot for the AAAI robot challenge. *AAAI Magazine* 42:2:51–72.

Wauchope, K. 1990. A tandem semantic interpreter for incremental parse selection. Technical Report NRL-9288, Naval Research Laboratory, Washington, D.C.

Wauchope, K. 1994. Eucalyptus: Integrating natural language input with a graphical user interface. Technical Report NRL/FR/5510-94-9711, Naval Research Laboratory, Washington, D.C.