International Journal of Social Robotics
DOI 10.1007/s12369-009-0032-4

Development and Validation of a Robust Speech Interface for Improved Human-Robot Interaction

Amin Atrash · Robert Kaplow · Julien Villemure · Robert West · Hiba Yamani · Joelle Pineau

Accepted: 17 September 2009

Abstract Robotics technology has made progress on a number of important issues in the last decade. However many challenges remain when it comes to the development of systems for human-robot interaction. This paper presents a case study featuring a robust dialogue interface for human-robot communication onboard an intelligent wheelchair. Underlying this interface is a sophisticated software architecture which allows the chair to perform real-time, robust tracking of the dialogue state, as well as select appropriate responses using rich probabilistic representations. The paper also examines the question of rigorous validation of complex human-robot interfaces by evaluating the proposed interface in the context of a standardized rehabilitation task domain.

Keywords Intelligent wheelchair · Dialogue management · Service robotics · Human-robot interaction

1 Introduction

For many people suffering from chronic mobility impairments, such as spinal cord injuries or multiple sclerosis, using a powered wheelchair to move around their environment can be difficult. According to a recent survey, 40% of patients found daily steering and maneuvering tasks to be difficult or impossible [7], and clinicians believe that between 61% and 91% of all wheelchair users would benefit from a smart wheelchair [29]. Such numbers suggest that the deployment of intelligent wheelchairs catering to those patients' needs could have a deep societal impact.

Over the last decade, robotics technology has made progress on a number of important issues pertaining to mobility. Many of these developments can be transfered to the design of intelligent wheelchairs. Yet many challenges remain—both technical and practical—when it comes to the development of the human-robot interaction components. A recent survey of the literature on smart wheelchairs suggests that while voice control has often been used to control smart wheelchairs, it remains difficult to implement successfully [28].

Our work addresses two main challenges pertaining to the development of voice-controlled assistive robots. First, we tackle the problem of robust processing of speech commands. In support of this goal, we propose a complete architecture for handling speech signals, which includes not only signal processing, but also syntactic and semantic processing, as well as probabilistic decision-making for response production. Many of these components have been exploited separately in human-machine speech interfaces, but to the best of our knowledge, this is the first system to incorporate all components in a coherent architecture for speech-based robot control.

Second, the paper tackles the issue of developing tools and standards for the formal testing of assistive robots. The use of standard testing has been common currency in some sub-tasks pertaining to human-robot interaction, most notably speech recognition. However few tools are available for the rigorous and standardized testing of

Corresponding author: J. Pineau
School of Computer Science, McGill University
McConnell Engineering Building, Rm 318
Montreal, PQ, CANADA
E-mail: jpineau@cs.mcgill.ca

fully integrated systems. Here we propose a novel methodology and environment for the standardized testing of smart wheelchairs. The procedure is inspired from one commonly used in the evaluation of conventional (non-intelligent) wheelchairs. We demonstrate, through careful user experiments, how it can be adapted to formally evaluate voice-controlled smart wheelchairs. Results described below show that the coupling of speech input, grammatical inference, and probabilistic decision-making provides a robust architecture to process natural interactions from untrained users.

The cognitive architecture described in this paper for handling speech interaction is appropriate for a wide spectrum of human-robot interaction tasks. As such, the work should be of interest to researchers working on a variety of social robot platforms, not only smart wheelchairs. The testing procedure we describe however is specifically targeted at the evaluation of smart wheelchairs. Nonetheless this case study may provide ideas and motivation for the development of standardized evaluation tools for other social robot interaction domains.

2 The SmartWheeler Platform

The SmartWheeler project aims at developing—in collaboration with engineers and rehabilitation clinicians—a prototype of a multi-functional intelligent wheelchair to assist individuals with mobility impairments in their daily locomotion, while minimizing physical and cognitive loads [23] .

Figure 1 shows a picture of the SmartWheeler platform. The intelligent wheelchair is built on top of a commercially available Sunrise Quickie Freestyle wheelchair. The intelligent sensing and computing components were designed and installed in-house at McGill University's Centre for Intelligent Machines. These include two (one forward-facing, one backward-facing) SICK laser range-finders, custom-made wheel odometers, a Lilliput 8" touch-sensitive LCD, a two-way voice interface, and an onboard 1.4 GHz Pentium M Processor. The laser range-finders and odometers are used for mapping, navigation and obstacle avoidance. The display, voice interface, and wheelchair joystick are the main modes of communication with the user. The onboard computer interfaces with the wheelchair's motor control board to provide autonomous navigational commands.

Fig. 1 The robotic wheelchair platform.

3 Cognitive Architecture

Figure 2 presents an overview of the software architecture underlying the communication modules. This includes the modules handling the various communication modalities (Speech Recognizer, Speech Synthesis, Visuo-Tactile Unit), a module handling grammatical parsing of the speech signal (Semantic Grammar), a core decision-making unit (Interaction Manager), a module to translate the decision into low-level output functions (Behavior Manager), and a module to handle the robot's low-level navigation components (Robot Control System).

Integration of the software architecture is done using the Acropolis framework [40]. In this framework, components are integrated as plug-ins that execute within the architecture. Plug-ins are defined in terms of what inputs they require and what outputs they produce; a so-called circuit file specifies which plug-ins are connected. This "loosely coupled" scheme allows for rapid prototyping through the switching of components on the fly. To connect components, Acropolis provides its own communication channel, making it easy and efficient to transfer data between components. Our current implementation includes only the components shown in Figure 2, however Acropolis is well suited to a variety of other robotic components, and is particularly suitable for quick testing of different configurations, since new components can be easily added, removed, or replaced.

Acropolis usually includes a plug-in to a standard Robot Control System. Good options for this module include the popular Player application [5], or the Carmen navigation toolkit [22]. The SmartWheeler's

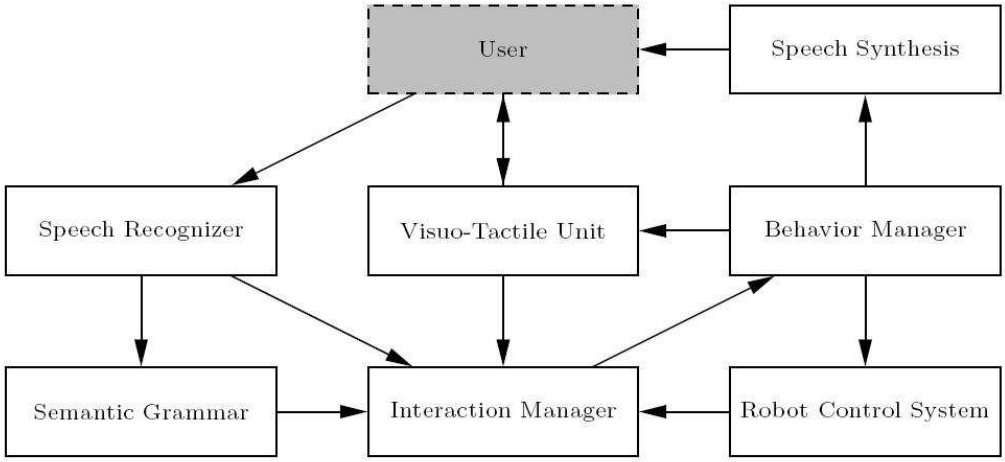

Fig. 2 Interaction Architecture

sensors and actuators can then be accessed directly through these navigation tools. As a result, the interaction architecture we present in this paper is relatively independent of the physical robot platform, and could be used for a wide variety of human-robot interaction systems.

3.1 Speech Recognition

A speech interface provides a comfortable and natural input modality for users with limited mobility. In general, speech requires little training, and is relatively high-bandwidth, thus allowing for rich communication between the robot and human. The performance of speech recognition systems are influenced by many aspects, including the vocabulary, language and acoustic models, speaking mode (isolated words vs. continuous speech), etc. Some of these aspects have to be taken into account when designing the speech interface.

Selecting a speech recognizer that performs well for the task at hand is important. To preserve flexibility in the development process, we considered two open source speech recognition systems: HTK [12] and CMU's Sphinx-4 [4]. Both of these systems are speaker-independent, continuous speech, recognition systems, which typically require less customization than commercial systems. Because customization is minimal, it is important that the system be pre-trained on a large speech corpus such that appropriate acoustic models can be precomputed. Such corpora usually falls under one of two categories: those developed for acoustic phonetic research and those developed for very specific tasks. Since SmartWheeler is still at an early stage of development, and domain-specific data is not available, we use a general purpose acoustic model, such as the Wall Street Journal Acoustic Model [36].

A small vocabulary makes speech recognition more accurate but requires the user to learn which words or phrases are allowed. And while our current focus is on building and validating an interaction platform for a specific set of tasks (defined below), we also want the user to be able to interact with the system in the same way s/he would, when interacting with any caregiver, and with very little prior training. Thus a fixed set of tasks is considered, but several possible commands for each task are allowed. For example, if a user wants to drive forward two meters, possible commands include (amongst others):

- Roll two meters forward
- Drive forward two meters.
- Roll forward two meters fast.
- Drive fast two meters forward.

The specific vocabulary allowed by the speech recognizer is extracted from a set of preliminary interactions recorded between users and the chair, based on the specific tasks selected. We provide more detail on our current implementation below.

Both Sphinx-4 and HTK allow the designer to specify a set of syntactic rules (or grammar) which specifies constraints on the ordering of words within a sentence. This grammar can be useful to enhance speech recognition quality by constraining the hypothesis space. In the SmartWheeler architecture, we have a separate module to handle both semantic and syntactic grammatical constraints (see the next section), therefore it suffices to use a very loose grammar in the speech recognition system.

The dictionary and grammar files used for both Sphinx-4 and HTK are available online [31].

We conducted a series of preliminary tests using both Sphinx-4 and HTK. Three male and one female university students in their early twenties recorded 179-183 commands each (723 speech commands in total). Subjects were presented with a script of commands, corresponding to each task, and instructed to read them in order.

We analyzed the results in terms of *substitutions* (when the speech recognizer fails to recognize a word and substitutes another incorrect word), *deletions* (words that were spoken but missed by the recognizer), *insertions* (words that were not spoken but added by the recognizer), *word error rate* (proportion of incorrectly recognized words, including substitutions, deletions and insertions), and *sentence error rate* (proportion of sentences in which one or more words are incorrectly recognized).

Both speech recognition packages showed equivalent performance across the board. Note that the results indicated rather high error rates. The mean sentence error rate was 45.2% when using Sphinx-4 and 46.7% when using HTK, while the mean word error rate was 16.1% with Sphinx-4 and 16.6% for HTK. This analysis suggests that performance of the two candidate speech recognition systems is equivalent. We chose to integrate Sphinx-4 onboard the SmartWheeler simply because it is slightly easier to handle in terms of software integration within Acropolis.

While the error rates are quite high, upon closer analysis the situation is not as discouraging as it would seem. Observed errors can be classified into one of three types: design errors, syntax errors, and semantic errors.

Design errors

Errors of this type are introduced because of errors in the design of the task vocabulary. For example, a number of substitutions occurred when Sphinx recognized METER as METERS. The task grammar within the speech recognizer was modified to avoid such minor errors.

Syntax errors

Errors of this type are introduced because the task grammar contains many ways to say the same thing. For example, when subject 1 said, ROLL BACK ONE METER, HTK recognized it as ROLL BACKWARD ONE METER. This is counted as one substitution error. However, even though the commands do not match, they do have the same semantic meaning in this particular task domain. The natural language processing module described in Section 3.2 is specifically designed to handle this type of error.

Semantic errors

Errors of this type are introduced when the speech recognition system fails to recognize the correct command. For example, when subject 3 said, DESCEND THE CURB, Sphinx recognized it as ASCEND THE CURB. In order to handle this type of error, it is necessary to reason about the environment and the user's intents. A more severe error of this type occurred when subject 3 said DRIVE SLOWLY TWO METERS BACK and Sphinx recognized ASCEND RIDGE. This error is harder to detect, but not impossible. Handling this type of error involves reasoning about the user's intentions and the robot's physical context. This is the main motivation for adopting a rich probabilistic framework for the Interaction Manager, as described in Section 3.7.

Evidence from these preliminary experiments strongly supports the integration of the diverse components included in the cognitive architecture.

3.2 Semantic Grammar

The output of the speech recognizer is processed by a natural language parser, which extracts syntactic and semantic information from the string of words. The primary role of the grammar is to constrain the output space of the speech interface. The framework we use for natural language parsing is that of Combinatory Categorical Grammars (CCG) [33], for which an open-source implementation is freely available [37].

It is worth noting that the grammatical analysis carried out in the NLP parser replaces the grammatical analysis that could be performed within the speech recognition system. The main advantage of using a separate NLP processing unit is that the CCG parser considers both syntactic and semantic constraints (as opposed to only considering syntactic constraints). Thus different words with identical semantic meaning can be mapped to each other. This can reduce the per-sentence error rate.

The semantic component of the parser is expressed in logical form, with logical predicates representing the meaning of the input sentence. The

set of these outputs forms the basis for the state space of the Interaction Manager. The set of possible outputs can grow quickly, depending on the space of logical predicates. The NLP parser keeps this to a manageable size by abstracting away all syntactical components for the later processing. The semantic logical form, which could take a hierarchical form (e.g. TWO METERS is the distance argument of ROLL TWO METERS FORWARD, and TWO is the number argument of the distance argument), is flattened into a fixed-size feature vector with pre-specified slots (e.g. ⟨Action: roll, DistanceUnit: meter, DistanceValue: 2, Direction: undefined, ...⟩). One can think of this representation as lying between the overly complex logical form common to full NLP systems, and the overly simple bag-of-words assumption used by many machine learning applications.

To conclude this section, it is worth emphasizing the advantages of using the CCG grammar as a separate module, over using the grammar offered in the Sphinx (or HTK) speech recognition system. First, the CCG grammar is more expressive than the simple Backus-Naur-Form context-free grammar framework used in Sphinx. Another important advantage of CCG is that it provides support for the construction of a logical-form semantic representation during the parsing process, which then forms the basic state representation for the Interaction Manager. Since the Interaction Manager's state space is based directly on the output space of the grammatical unit, substantial compression in the state space (and decision space) of the Interaction Manager is achieved through the application of the CCG parser's syntactic and semantic constraints. This significantly improves the scalability of the interaction system.

In cases where the CCG parser does not return a valid parse, we can still recuperate the set of words produced by the speech recognition system and pass them directly to the Interaction Manager to be processed as a bag-of-words, without any syntactic or semantic structure. This is the standard approach in dialogue management systems that do not include grammatical parsing [6, 27]. In some cases, this can help guide the selection of appropriate clarification queries can be preferable to having the speech recognition system go to unnecessary lengths to fit the observed speech signal into a valid (but incorrect) parse.

3.3 Speech Synthesis

In order to provide speech feedback, the Festival [9] speech synthesis system is used. This system is freely available, and does not require customization, therefore we do not discuss it further. The list of speech output sentences is set by hand and stored in the Behavior Manager.

3.4 Visuo-Tactile Unit

A graphical user interface (GUI) complements the speech interface by providing immediate visual feedback to the user, as well as a secondary mode of input through the tactile interface. Note that the screen-based interaction and speech-based interaction modes are thoroughly complementary. At any point during the interaction, the user can enter commands either through the speech interface, or through the touch-sensitive display. Additionally, there is in fact a third mode of input, which is to select the buttons of the visuo-tactile display using a joystick (or equivalent device). The screen changes reactively whenever instructions are issued using any of the three input modalities.

The primary motivation behind the GUI design is to provide an alternative to the speech interface for users who are not able to reliably control the wheelchair using voice. Some people may have disabilities which interfere with their vocal abilities; others may find the speech interface too error-prone and prefer more reliable alternatives. The relative level of reliability of course will be affected by the user's fine motor control.

Another motivation for the GUI design is to provide context and support to facilitate the use of the speech interface. The display can be used to inform of the current state of the wheelchair (e.g. moving, stopped), as well as the set of commands that can be used in that particular situation. The user is always free to voice commands that do not appear on the screen (this can lead to "short-cuts" through the menu system).

We do not formally evaluate the GUI design in experiments reported below, therefore we do not describe this module in detail. Design and validation of this component will be the subject of future work, since they require a substantial involvement from the disabled user population.

3.5 Behavior Manager

The role of the behavior manager is to provide an interface layer between the high-level decision-making and the low-level robot control. As explained below, the Interaction Manager chooses high-level action in response to the received processed sensory input. These actions are either robot control actions, such as movement or hardware configuration, or response actions to be communicated to the user. The Interaction Manager passes the selected action to the Behavior Manager, which then translates the high-level action into lower-level robot specific instructions. This allows the hardware specific aspects to be abstracted away from the interaction and decision-making modules. The Behavior Manager is implemented as a simple look-up table, in which the decomposition of a high-level action is detailed as a sequence of low-level commands for the appropriate modules (speech generation and/or robot navigation).

3.6 Robot Control System

The Robot Control System can be implemented using different publicly available software packages, for example the popular Player application [5] or the Carmen robot navigation toolkit [22]. We have used both throughout our preliminary experiments. The goal of this component is to handle tasks such as mapping, localization and path planning. We do not discuss this component further as it is somewhat orthogonal to the main focus of this paper.

3.7 Interaction Manager

The Interaction Manager acts as the core decision-making unit in the robot architecture. This module is ultimately responsible for selecting the behavior of the robot throughout the interaction with the user.

In this context, the Interaction Manager can be seen as an Input-Ouput device, where information about the world is received via the grammar system and the low-level robot navigation system. The unit then outputs actions in the form of speech and display responses, or issuing of control commands to the navigation unit. These actions are processed through the Behavior manager, to extract a pre-set sequence of low-level operations, before being sent to the respective modules (speech synthesis, visuo-tactile unit, robot control system).

The goal of the Interaction Manager is to provide a robust decision-making mechanism capable of handling the complexity of the environment. This is a challenging goal due to the high degree of noise in the environment. While the semantic grammar can help handle some of the noise, even properly transcribed speech can contain ambiguities. Accounting for the set of possible outcomes can be crucial in providing robust action selection. Therefore we favor a probabilistic approach to decision-making.

The Partially Observable Markov Decision Process (POMDP) paradigm has been shown to be a powerful tool for modeling a wide range of robot-related applications featuring uncertainty, including robot navigation [18], dialogue management [30, 27, 38, 6], and behavior tracking [11]. One of the advantages of the POMDP model is its ability to capture the idea of *partial observability*, namely that the state of the world cannot be directly observed, but instead must be inferred through noisy observations [32]. In the case of our Interaction Manager, the user's intention must be inferred though observations coming from verbal commands, as well as contextual information inferred through the physical state of the robot in its environment.

Each state in the POMDP represents a discrete user intention, such as moving forward, turning the chair, or reaching a specific target. The states are defined using a pre-defined set of semantic slots, and a finite domain for each slot.

The output of the Interaction Manager is defined by a set of actions. Because each state represents a different user intention, there is one "correct" action for each state. This includes robot control actions and interaction response actions. If the agent selects and executes the correct action, a positive reward is given. If the incorrect action is selected, a high cost is inflicted. In addition, there is a set of query actions, in the form of clarification questions, which can be emitted in situations that warrant additional information. All query actions incur a small cost. The goal of the decision-making engine is to select actions so as to maximize this reward function.

Observations occur when there is input from the sensor-processing modules. Most of the time, these appear as an output of the grammatical parser, and consist of assignments to the various semantic slots. Given the observation, the goal is to determine the likelihood of each state, s_i, given this assignment, $z = (CommandValue = v_1, CommandType = v_2,$

$DirectionValue = v_3, ...)$, written as $z = (v_1, v_2, v_3, ...)$ for readability. We assume that assignments of values to the slots is independent:

$$P(s_i|v_1, v_2, v_3, ...) = \frac{P(v_1, v_2, v_3, ...|s_i)P(s_i)}{P(v_1, v_2, v_3, ...)} \quad (1)$$
$$= \frac{P(v_1|s_i)P(v_2|s_i)P(v_3|s_i)...P(s_i)}{P(v_1)P(v_2)P(v_3)...}$$

These values, $P(v_j|s_i), P(v_j)$, and $P(s_i)$ can be calculated from collected data.

In cases where there was no valid parse, the phrase is sent directly from the Speech Recognizer to the Interaction Manager as a bag-of-words. The state probability update is done in similar fashion, but considering the probability of each word, given the state. Again, these can be learned from data.

Ideally, the transition from state-to-state should be estimated from training data, to reflect patterns in daily activities. In cases where such data is not available, one can assume a uniform transition to any other state following execution of a non-query action (i.e. we don't know what task the user is going to tackle next) and assume the system remains in the same state (with slight noise) following execution of a query action.

Once the model is created, a POMDP solving algorithm is applied to determine a proper policy. A policy is a mapping from distributions over states (as defined in Equation 1) to actions. A typical policy might be as follows: When the system is confident about the user's intention, the system executes the corresponding action. If there is some degree of ambiguity, such as the system believing the action is a MOVE command, but lacking the parameters, the system can then execute a specific query, such as REQUEST MOVE PARAMETERS. If there is sufficient noise and the system is unsure of the state, it can execute the more general REPEAT query. This policy is optimized using dynamic programming methods based on the transition, observation, and reward parameters. As the cost of queries decreases, or the penalty for an incorrect action increases, the system will become more cautious and inclined to present multiple queries before committing to an action. Several techniques exist for determining a policy from a model. We use a standard point-based approximation technique [24].

4 A Standardized Evaluation Tool for Smart Wheelchairs

In his survey of smart wheelchairs, Simpson [28] comments that: *"While there has been a significant amount of effort devoted to the development of smart wheelchairs, scant attention has been paid to evaluating their performance. (...) Furthermore, no smart wheelchair has been subjected to a rigorous, controlled evaluation"*. His comments are made mostly in the context of highlighting the need for better experiments with the target population, in real-world settings. However, before we—as a community—are in a position to achieve this full goal, a necessary step is to develop the tools and methods for achieving such rigorous, controlled evaluation. This section describes one such tool.

4.1 Target population

The target population for our smart wheelchair platform is those users who require powered wheelchair to get around their environment on a daily basis, yet find the use of a wheelchair problematic, whether due to fatigue, limited motor skills, or sensory impairment. This can include individuals suffering from spinal cord injuries, multiple sclerosis, or other mobility disorders.

The experiment described in the latter section of this paper do not include disabled users; all results reported below were obtained with healthy subjects. The primary goal of the experiment presented below is to serve as a pilot study for the project. This is an important step towards acquiring ethical approval for the experiments with the disabled population.

Before conducting the pilot study, we consulted at length with clinical experts to define the selection criteria for the target population which will be included in later phases of experimentation. This selection criteria includes: a reduced mobility component (i.e. verifying the subject's need for a motorized wheelchair), a communication fluency component (i.e. verifying the subject's ability to use a two-way speech interface), as well as other criteria ensuring the safety and well-being of the test subjects. In the context of the pilot study, the reduced mobility criterion was ignored, but all other criteria were applied without modification. Given that our primary goal in this paper is to evaluate the effectiveness and robustness of the human-robot interface, and in particular robust speech interaction, we believe that the results obtained in the pilot study with healthy subjects will be reasonably indicative of results for the target population, assuming similar communication abilities.

4.2 A Standardized Rehabilitation Environment

The long-term goal of this project is to increase the autonomy and safety of individuals with severe mobility impairments by developing a robotic wheelchair that is adapted to their needs. This is a reasonably ambitious goal, which can entail a wide spectrum of tasks and activities.

While the long-term goal is to deploy our platform in natural indoor/outdoor living environments adapted for wheelchair users, there is a need in the earlier stages of the project to formally assess the effectiveness and safety of the platform in a more controlled task domain and environment. In seeking such a testing methodology, we have come across a standard wheelchair training protocol called the *Wheelchair Skills Test* (WST). The WST was developed to assess and train wheelchair users through a representative set of wheelchair skills [15]. Extensive information about the test can be found on the WST website[1]. The test is currently being used in clinical settings to identify skills that should be addressed during training, as well as to compare a subject's performance before and after rehabilitation.

There are many reasons why we believe this testing methodology is useful for the validation of smart wheelchairs. The WST includes 39 skills, divided into 18 skills groups, and representing multiple levels of difficulty ranging from simple tasks (such as moving forward short distances) to more complex tasks (such as performing turning manoeuvres in constrained spaces). The set of skills can be thought of as an "obstacle-course" for wheelchairs, and is considered representative for general wheelchair performance. The assumption is that a person doing well on the 39 tasks included in the WST can be considered a skilled wheelchair user because the abilities tested are sufficiently rich to resemble situations encountered on a daily basis. The choice of tasks is based on realistic scenarios, but is still standardized enough to allow for precise performance measurements. As one would expect, most tasks test navigation skills (e.g. Roll forward 10 m in 30 s, Get over 15-cm pothole), but there are some other actions as well, e.g. those concerning the wheelchair configuration, like Controls recline function. Accomplishing all tasks takes roughly 30 minutes [16]. The test does require some environment infrastructure (e.g. a ramp, a few walls) but the space requirements are not excessive and typical of the space required for standard mobile robotics research.

Other wheelchair skills tests have been proposed in the occupational therapy literature. See [14] and [26] for comprehensive overviews of existing tests. Kilkens *et al.* concluded that out of the 24 tests they reviewed, only the Wheelchair Skills Test has been *"adequately tested on both validity and reliability"*. It is also one of the few tests which was designed for powered wheelchairs. Furthermore, it has not been designed for a specific target group (e.g. stroke patients), but for wheelchair users in general. This is a useful aspect since our aim in designing the SmartWheeler platform is to improve mobility for a large spectrum of mobility-impaired individuals.

For our use of the WST, it is particularly interesting to note that a version of this test was explicitly conceived to provide a means of evaluating the wheelchair+user+caregiver as a team. The goal is not to rate the performance of the wheelchair user, but rather that of this team. In the case of an intelligent system, we view the added robot technology as playing a role similar to that of a caregiver and we use the WST in the version 'powered wheelchair with caregiver' (WST-P/CG in [15]). The WST manual specifies that a caregiver need not be a human: *"An animal (e.g. a service dog) that assists with the performance of a skill is considered a caregiver, not an aid."* [15] We are extending this notion even further by introducing the intelligent system as a caregiver. Although the WST manual does not explicitly define a caregiver, this is justified because the purpose of our intelligent software system is in fact to cooperate with the wheelchair user in order to accomplish the target activities.

Finally, it is worth emphasizing that the WST evaluates skill proficiency, as well as safety. The WST includes a formal evaluation protocol, whereby performance on each of the tasks is graded in terms of these two criteria in a binary manner: a person either passes or fails a task, and he/she does so either in a safe or in an unsafe way. The pass/fail grading method makes the evaluation simple and as objective as possible. This is reflected in the high test-retest, intra- and inter-rater reliabilities achieved by the WST [16,17]. Because the set of tasks and the testing environment is precisely defined, it is easier to provide strong guarantees regarding the safety and security of the wheelchair in this constrained domain, compared to natural living environments. Thus it is also easier to en-

[1] `http://www.wheelchairskillsprogram.ca`

sure we meet ethical requirements for testing with the target population.

4.3 Customization of the Cognitive Architecture for the WST

The description of the cognitive architecture in Section 3 was done in reasonably general terms to emphasize that it is appropriate for a large spectrum of human-robot interaction tasks. In this section we outline a few of the design decisions that were made to deploy this architecture onboard our robot platform and in the context of the Wheelchair Skills Test.

The vocabulary used by the speech recognizer was designed to include 61 words, which were determined by going through sample interactions between users and the chair, based on the specific tasks of the WST. The dictionary and grammar files used for both Sphinx and HTK are available online [31]. Throughout our experiments, the speech recognition system was used in press-to-speak mode, requiring the subjects to press a button while speaking to the robot. This could be an obstacle for severely disabled individuals, however given that one of our selection criteria is that the person currently uses a powered wheelchair (which s/he controls through a joystick or other pressure sensor), this poses no problem at this time in our investigations.

Regarding the design of the logical rules in the semantic grammar, obviously the goal is not to equip the robot with a full-fledged English grammar at this stage of the project. The current grammar is based on the tasks of the Wheelchair Skills Test. Our CCG grammar file is also available online [31]. In cases where an input sentence has multiple correct parses, we have to make a choice on which parse to pass on to the Interaction Manager. Currently we pick the first parse from the list. In the future we will investigate methods to learn from experience which parse is most informative, or consider extending the Interaction Manager such that it accepts multiple parses.

To realize the interface for the WST, the set of semantic slots produced by the grammar and used as the state representation for the interaction manager are presented in Table 1. In reality, we use a slightly larger set, generated based on data. Each state represents one set of assignments to each of the semantic slots. Typically, a state has 3 to 6 semantic slots assigned, with the remainder being set to null.

Semantic Slot	Possible Value Assignments
CommandValue	move, switch, activate, deactivate, select, set, drive, turn, NULL
CommandType	move-action, hardware-action, config-action, NULL
DirectionValue	away, back, forward, backward, right, left, NULL
DirectionType	direction, vector-direction, angle-direction, NULL
PatientType	controller, speed, drive-mode, seat, motor, NULL
PatientValue	on, off, fast, medium, slow, cautious, indoor, outdoor, NULL

Table 1 Semantic Slots and Possible Value Assignments

The POMDP domain file implementing the Interaction Manager is available online [31]. Statistics for the observation probabilities in the case of bag-of-words were calculated from data gathered during our preliminary study of the speech recognition system (see Section 3.1). We assume uniform transition from state-to-state after executing any non-query action, and stationary transitions (with slight noise) after executing query actions. This is adequate to capture the Wheelchair Skills Test, where the sequence of tasks is not meaningful.

In addition to the evaluation metrics included in the WST, to better characterize our interface, we consider other metrics such as: the speech recognition accuracy, the proportion of sentences that are correctly parsed, and the number of correct action choices by the Interaction Manager.

5 Empirical Evaluation

We conducted experiments to evaluate the performance of our cognitive architecture in the context of the WST task domain. The evaluation was conducted in our laboratory. Seven healthy subjects were involved. Five male and two female university students in their early twenties were recruited for the evaluation. All were healthy subjects, without physical disabilities. None of the subjects were directly involved in the development of the SmartWheeler robot.

Subjects were given a brief (5 minutes) introduction to the robot, including a high-level description of the Wheelchair Skills Test task domain, and interface modalities. Unlike in the preliminary evaluation (Section 3.1), where subjects were given a script of commands, in this latter evaluation subjects were only give a description of each task (as precisely scripted in the WST),

and chose themselves the words to communicate with the robot. While the wheelchair did not move during these experiments (such that the performance of the interface could be studied independently from any navigation issue), the users were provided with feedback about the robot's (simulated) current state through the screen-based interface.

Throughout the interactions, the time required to compute the robot's response, for each interaction, was on the order of 1 second (and in some cases much less). In general, most of the computation time is taken by the speech recognition; the grammatical analysis, POMDP decision-making, and behavior selection are comparatively much faster.

Most of the subjects tackled 21 of the tasks in the Wheelchair Skills Test (version WST-P/CG), except subject 3, who tackled 22 tasks (due to a protocol error on the part of the tester) and subject 4, who did 19 (due to recording issues). Tasks were presented to the test subjects using instructions specified in the WST manual. E.g. NAVIGATE YOUR CHAIR OVER THE FINISH LINE ONE METER AHEAD OF YOU, PUT YOUR CHAIR INTO THE FAST SPEED SETTING, and TURN THE WHEELCHAIR TO YOUR RIGHT ABOUT 90 DEGREES. The phrasing used to present tasks to the test subjects was such as to allow flexibility in the actual phrases used to issue the commands to the chair, as opposed to the subjects simply repeating the task description.

Table 2 shows results for this set of test subjects. First, we notice that the speech recognition rates are very poor (much worse than the preliminary study described in Section 3.1). This is explained by the fact that in this case, subjects were free to formulate sentences as they wished, rather than reading off a script. Furthermore, they were not even informed of the vocabulary accepted by the speech recognizer. The number of sentences parsed by the CCG module is also quite low, though this is not necessarily a problem, as we demonstrate through an example below. The number of correctly identified tasks shows substantial variance between subjects, with five of the subjects achieving high performance on the overall task, despite the poor recognition accuracy.

The number of speech commands per tasks captures the frequency of queries issued to help resolve the task at hand. If we consider subject 7, 2.10 commands were required for each task on average, meaning one command for the task itself and with an additional 1.10 queries (on average) for each task. Subject 3 required significantly fewer queries per tasks, with only one query for every third action, due to a higher recognition accuracy. In most cases, whenever the task was incorrectly identified, a very similar task was selected. Prior to starting the experiments, subjects were instructed that if at any point in time they were unsatisfied with the interface's behavior (e.g. wrong task was selected, or repetitive queries), they could use a specific command (STOP) to end the task and move on to the next one. This rarely happened, and results for these trials are included as incorrect tasks in column 5 of Table 2. More flexible methods of error recovery will be investigated in future phases of the project.

We now consider a few illustrative interactions transcribed from the experiment.

Example 1: Figure 3 shows a recorded dialogue sequence. This example shows the flexibility that is afforded by the modularity of our architecture. In the first steps of the interaction, the sentence is correctly parsed by the NLP grammar. The Interaction Manager nonetheless chooses a clarification query because the information provided is insufficient to select a task. In the latter steps, the user provides very succinct answers, which cannot be parsed by the grammar, yet the Interaction Manager is able to use these (under the bag-of-word assumption) to correctly update its state distribution and eventually produce the correct action.

Example 2: Looking at the subjects who had low task correctness rates yields useful insight into weaknesses of the current system. Test subject 5 was extremely verbose while issuing commands, saying for example (Figure 4) LET'S MOVE TOWARDS THE WALL AND END UP PARALLEL TO THE WALL in an attempt to align the wheelchair to the wall, whereas most other subjects simply said ALIGN THE CHAIR TO THE WALL. The system was often able to infer the task based on informative words such as WALL, though in some cases the system was unable to compensate. The opposite effect was observed in test subject 2, who used extremely terse language, issuing commands such as BACK and FAST to capture tasks such as MOVE BACKWARDS ONE METER and MOVE FORWARD QUICKLY.

For some subjects, the task recognition rate is high, but the number of queries used is also relatively high. This generally indicates poor speech recognition. In our study, subjects 1 and 6 are non-native English speakers, with residual accents, for

Subject Id	Word Err. Rate	Sentence Err. Rate	% Sentence Parsed	% Task Correct	# Commands per Task
1	99.2	95.8	3	76	1.81
2	51.3	74.5	29	43	1.75
3	39.4	67.5	38	95	1.38
4	51.2	63.6	34	79	1.68
5	93.4	97.3	0	29	1.62
6	62.0	71.4	14	76	2.43
7	48.9	54.9	41	76	2.10
Mean	63.6	75.0	22	67	1.82

Table 2 Results from Test Subject Interaction.

```
Command 1:
<SentenceRecognized value=‘‘true" />
<Phrase value=‘‘move backward" />
<AllPhrases value=‘‘[move backward]" />
Actual: ‘‘Move backward"
<Action value=‘‘move" type=‘‘move-action" />
<SpeechAct value=‘‘command" />
<Direction value=‘‘backward"
             type=‘‘vector-direction" />
Best Action: MOVE_QUERY

Command 2:
<SentenceRecognized value=‘‘false" />
<Phrase value=‘‘backward" />
<AllPhrases value=‘‘[backward]" />
Actual: ‘‘Backward"
Best Action: MOVE_QUERY

Command 3:
<SentenceRecognized value=‘‘false" />
<Phrase value=‘‘ "/>
<AllPhrases value=‘‘[backward set]" />
Actual: ‘‘Backwards"
Best Action: drive backward
```

Fig. 3 A typical sequence including queries and goal task. For each speech command, lines 2 and 3 indicate whether the sentence could be parsed. Line 4 is the output of the speech recognizer. Line 5 is a hand-labeled transcription. The last line shows the action selected by the Interaction Manager.

```
Command 1:
<SentenceRecognized value=‘‘false" />
<Phrase value=‘‘ " />
<AllPhrases value=‘‘[the move to wall to wall and
            and and up veer overcome wall on]" />
Actual: ‘‘Let’s move towards the wall and end up
            parallel to the wall"
Best Action: 21 : align to wall
```

Fig. 4 Example of verbose user input. Lines 2 and 3 indicate that the sentence could not be parsed by the grammar. Line 4 is the output of the speech recognizer. Line 5 ("Actual:") is a hand-labeled transcription added for clarification. Line 6 shows the action selected by the Interaction Manager.

which the speech recognition system is not trained. Subject 7 also had poor recognition accuracy, possibly due to a high speech volume. In each case, speaker-customization could substantially reduce the number of queries. It is nonetheless encouraging to see that even though the recognition accuracy was low, the number of correct tasks was substantially better.

6 Related Work on Intelligent Wheelchair Platforms

Wheelchairs are a challenging platform for intelligent agents and are being explored by many groups as a way to help disabled and elderly individuals. There is an excellent, and fairly recent, review of intelligent wheelchair platforms [28], which provides a summary of research in this area over the last 20 years. It classifies existing systems in terms of their form factor, input methods, onboard sensors, and control software.

Much of the most recent work on intelligent wheelchairs focuses on the navigation behavior of the chair, for instance providing smooth motion to improve user comfort and investigating sophisticated representations of the physical environment [10, 2]. We do not review this work in detail as it tackles issues that are orthogonal to the focus of this paper.

Many of the earlier intelligent wheelchairs prototypes used a voice interface, as does the SmartWheeler platform, to allow the user to provide commands to the robot [25,19,13,20,1,21,3]. Earlier systems were often prone to errors in the speech recognition, which was an impediment to robust interaction. It is worth noting that most of these systems preceded the latest statistical methods for language understanding and dialogue management. It is an important contribution of this paper to

show how this recent technology can be used to improve the robustness of the interaction.

We are aware of one instance of an autonomous wheelchair which has a focus on human-robot interaction and uses probabilistic decision-making in the dialogue manager [6], similar to the approach proposed here. However, this system is limited to speech-to-speech interactions, and does not include grammatical inference. It also has not yet been validated in the context of clinically-relevant task domains.

More recently, researchers have investigated the development of methods of simplified interfaces for human-robot communication. This includes for example mounting a robotic arm on a wheelchair to provide assistance for patients who have difficulty manipulating objects in their environment [35]. Another interesting example is the work on providing a concise communication protocol for entering text commands into the wheelchair using joysticks or touch-pads [39]. Finally, other researchers have investigated the use of EEG signals to control a powered wheelchair [8]. Such work is useful for patients with limited vocal abilities, but may be less preferred due to the lower communication bandwidth and the longer training time.

There is a large body of work on rehabilitation robotics, whereby robots aim to provide instructions and/or physical assistance to help individuals with disabilities to regain some of their mobility. Some of this work considers ideas of robustness and adaptation, by providing learning mechanisms to improve the robot's performance [34]. Such methods may be useful at a later stage in our work, even though the application is fundamentally different.

7 Conclusion

The aims of this paper are two-fold. First, we present a cognitive architecture for voice-driven human-robot interaction. The emphasis is on achieving robust interaction through a number of key components: Speech Recognition, Natural Language Processing, Interaction Manager, and Behavior Manager. All of these components have been used in human-robot interfaces previously, but not altogether. To the best of our knowledge, this is the first system to integrate both grammatical parsing and probabilistic decision-making. This is an important step towards achieving robust, yet flexible, human-robot interaction in complex task domain. The primary interface modality has been speech, but the architecture is well equipped to handle input from other modalities, such as a joystick or touch-sensitive display. This is the topic of ongoing work.

The second important aspect of this work is the emphasis on evaluating the robot platform in the context of a standardized, clinically relevant, domain. The lack of formal evaluation has been identified as one of the key challenges for researchers developing intelligent wheelchair platforms. In this paper, we argue that the Wheelchair Skills Test domain is a useful and well-specified instrument for such a purpose. While its applicability is limited to wheelchair-type devices, many of its characteristics should provide inspiration for developing similar tools for other assistive robotic platforms. It is important to remember that evaluation on standard environments is an important step towards demonstrating safety and security of robot platforms, and this is especially crucial for human-robot applications as we prepare to move towards more natural living environments.

One of the risks of focusing evaluation on highly standardized test environment is that it may lead to over-specialization of the robot interface and platform to this precise environment. It is certainly the case that certain components (e.g. the vocabulary of the Speech Recognizer, the logical rules in the grammar, and the state and action sets used in the Interaction Manager) were designed to be task specific. Nonetheless the overall cognitive architecture we proposed could be re-used without modification for a wide array of task domains. In fact, we have used the architecture and platform for a number of experiments unrelated to those described in this paper without much effort.

In conclusion, the work presented here offers useful insights into the design and evaluation of voice-controlled interfaces for intelligent robots. The main focus is on the integration of components such as grammatical parsing and probabilistic decision-making, which offer robust processing of the interface data. While the success rate is still far from perfect, we show that the integration of these components contributes to improving the robustness of the system, compared to using only speech recognition as an input to select behaviors. An important next step is to validate these results with the target population. We anticipate that the results will be similar, especially for subjects whose communication abilities are intact, despite having mobility impairments.

In the longer term, we also aim to improve the task success rate, as well as tackle a larger set of

tasks, such that the system can be effective in a rich set of situations. One important direction for improving the rate of success is in customizing the interface for each user. This can be achieved in a number of ways, for example by exploiting a user-specific acoustic model for the speech recognition, or by learning the grammatical inference rules automatically from user data, as well as by adapting the reward function of the POMDP-based interaction manager based on user preferences. There exist good algorithmic methods for realizing these improvements, however these are often data-expensive. There are important research opportunities in finding ways to achieve user customization without requiring excessive amounts of data.

Acknowledgements The authors would like to thank its research partners, in particular Paul Cohen from École Polytechnique de Montréal, Robert Forget of the Université de Montréal, and the clinicians at the Centres de réadaptation Lucie-Bruneau and Constance-Lethbridge. The authors also acknowledge financial support by the Natural Sciences and Engineering Council Canada (NSERC), as well as the Fonds Québécois de la recherche sur la nature et les technologies (FQRNT).

References

1. Amori, R.D.: Vocomotionan intelligent voice-control sys-tem for powered wheelchairs. In: Proceedings of the RESNA 1992 Annual Conference, pp. 421–423 (1992)
2. Beeson, P., MacMahon, M., Modayil, J., Murarka, A., Kuipers, B., Stankiewicz, B.: Integrating multiple representations of spaitla knowledge for mapping, navigation, and communcation. In: AAAI Spring Symposium Series 2007, Interaction Challenges for Intelligent Assistants, AAAI Technical Report SS-07-04 (2007)
3. Clark, J.A., Roemer, R.B.: Voice controlled wheelchair. Arch Phys Med Rehabil **58**(4), 169–175. (1977)
4. CMU: Sphinx-4 (2004). `http://cmusphinx.sourceforge.net/sphinx4/`
5. Collett, T.H.J., MacDonald, B.A., Gerkey, B.P.: Player 2.0: Toward a practical robot programming framework. In: Proceedings of the Australasian Conference on Robotics and Automation (2005)
6. Doshi, F., Roy, N.: Efficient model learning for dialog management. In: International Conference on Human-Robot Interaction, pp. 65–72 (2007). DOI http://doi.acm.org/10.1145/1228716.1228726
7. Fehr, L., Langbein, E., Skaar, S.B.: Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. J. of Rehabilitation Research & Development **37**, 353–360 (2000)
8. Felzer, T., Freisleben, B.: Hawcos: the "hands-free" wheelchair control system. In: Proceedings of the fifth international ACM conference on Assistive technologies (2002)
9. Festival: Festival speech synthesis system (2004). `http://www.cstr.ed.ac.uk/projects/festival/`
10. Gulati, S., Kuipers, B.: High performance control for graceful motion of an intelligent wheelchair. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2008)
11. Hoey, J., Poupart, P., Boutilier, C., Mihailidis, A.: POMDP models for assistive technology. In: AAAI Fall Symposium on Caring Machines: AI in Eldercare (2005)
12. HTK: HTK 3.4 (2006). `http://htk.eng.cam.ac.uk/`
13. Katevas, N.I., Sgouros, N.M., Tzafestas, S.G., Papakonstantinou, G., Beattie, P., Bishop, J.M., Tsanakas, P., Koutsouris, D.: The autonomous mobile robot senario: A sensor-aided intelligent navigation system for powered wheelchairs. In: IEEE Robotics and Automation Magazine, pp. 60–70 (1997)
14. Kilkens, O.J.E., Post, M.W.M., Dallmeijer, A.J., Seelen, H.A.M., van der Woude, L.H.V.: Wheelchair skills tests: A systematic review. Clinical Rehabilitation **17**, 418–430 (2003)
15. Kirby, R.L.: Wheelchair Skills Program (WSP), Version 4.1. Wheelchair Skills Test (WST) Manual (2007). `http://www.wheelchairskillsprogram.ca/eng/4.1/WSTManual_Version_4.1.pdf` (accessed 04/27/2008)
16. Kirby, R.L., Dupuis, D.J., MacPhee, A.H., Coolen, A.L., Smith, C., Best, K.L., Newton, A.M., Mountain, A.D., MacLeod, D.A., Bonaparte, J.P.: The Wheelchair Skills Test (version 2.4): Measurement properties. Arch. Phys. Med. Rehabil. **85**, 794–804 (2004)
17. Kirby, R.L., Swuste, J., Dupuis, D.J., MacLeod, D.A., Monroe, R.: The Wheelchair Skills Test: A pilot study of a new outcome measure. Arch. Phys. Med. Rehabil. **83**, 10–18 (2002)
18. Koenig, S., Simmons, R.: Unsupervised learning of probabilistic models for robot navigation. In: International Conference on Robotics and Automation (1996)
19. Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y., Borenstein, J.: The navchair assistive wheelchair navigation system. IEEE Transactions on Rehabilitation Engineering **7**(4), 443–451 (1999)
20. McGuire, W.R.: Voice operated wheelchair using digital signal processing technology. In: Proceedings of the 22nd Annual International Conference on Assistive Technology for People with Disabilities (RESNA), pp. 364–366 (1999)
21. Miller, G.E., Brown, T.E., Randolph, W.R.: Voice controller for wheelchairs. Med Biol Eng Comput. **23**(6), 597–600 (1985)
22. Montemerlo, M., Roy, N., Thrun, S.: Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. pp 2436–2441 (2003)
23. Pineau, J., Atrash, A.: SmartWheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction. In: AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics, pp. 59–64 (2007)
24. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for POMDPs. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1025–1032 (2003)
25. Pires, G., Nunes, U.: A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. Journal of Intelligent Robotic Systems **34**(3), 301–314 (2002)
26. Routhier, F., Vincent, C., Desrosiers, J., Nadeau, S.: Mobility of wheelchair users: A proposed performance assessment framework. Disability & Rehabilitation **25**, 19–34 (2003)

27. Roy, N., Pineau, J., Thrun, S.: Spoken dialog management using probabilistic reasoning. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL) (2000)
28. Simpson, R.C.: Smart wheelchairs: A literature review. Journal of Rehabilitation Research & Development **42**(4), 423–436 (2005)
29. Simpson, R.C., LoPresti, E.F., Cooper, R.A.: How many people would benefit from a smart wheelchair? J. of Rehabilitation Research & Development **45**, 53–72 (2008)
30. Singh, S., Litman, D., Kearns, M., Walker, M.: Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. Journal of Artificial Intelligence Research (JAIR) **16**, 105–133 (2002)
31. SmartWheeler: (2008). `http://www.cs.mcgill.ca/~smartwheeler/data.php`
32. Sondik, E.: The optimal control of partially observable Markov decision processes. Ph.D. thesis, Stanford University (1971)
33. Steedman, M., Baldridge, J.: Combinatory Categorial Grammar. Blackwell (2005). `ftp://ftp.cogsci.ed.ac.uk/pub/steedman/ccg/manifesto.pdf`
34. Tapus, A., Tapus, C., Mataric, M.: Hands-off therapist robot behavior adaptation to user personality for post-stroke rehabilitation therapy. In: International Conference on Robotics and Automation, pp. 1547–1553 (2007)
35. Tsui, K., Yanco, H.: Simplifying wheelchair mounted robotic arm control with a visual interface. In: AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robots, pp. 97–102 (2007)
36. Vertanen, K.: HTK wall street journal acoustic models (2006). URL `http://www.inference.phy.cam.ac.uk/kv227/htk/acoustic_models.html`
37. White, M.: OpenCCG: The OpenNLP CCG Library (2001). `http://openccg.sourceforge.net`
38. Williams, J., Poupart, P., Young, S.: Partially observable Markov decision processes with continuous observations for dialogue management. In: SigDial Workshop on Discourse and Dialogue, pp. 393–422 (2005)
39. Wobbrock, J.O., Myers, B.A., Aung, H.H., LoPresti, E.F.: Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads. In: Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility (2004)
40. Zalzal, V.: Acropolis: Une architecture logicielle évolutive pour plate-formes mobiles autonomes: Principes de fonctionnement. Tech. rep., École Polytechnique de Montréal (2004)

Amin Atrash is a Ph.D. candidate in Computer Science at McGill University. He received his B.Sc. and M.Sc. in Computer Science at the Georgia Institute of Technology. His research interests focus on planning under uncertainty and human-robot interaction.

Robert Kaplow is an M.Sc. candidate in Computer Science at McGill University, where he previously completed is B.Sc. His research interests include mobile robot navigation, probabilistic planning and reinforcement learning.

Julien Villemure received his B.Sc. in Computer Science from McGill University in 2008, where he is now an M.Sc. candidate. His research focuses on developing and validating software architectures for human-robot interaction systems.

Robert West received received a Diplom in Computer Science from the Technische Universität München, then joined McGill's School of Computer Science, where he is an M.Sc. candidate. His research interests are in natural language processing, and in developing novel methods for inferring semantic distances between concepts.

Hiba Yamani completed her M.Sc. in Computer Science at McGill University in 2009. She is now an application developer at Nuance Communications.

Joelle Pineau is Assistant Professor of Computer Science at McGill University. She received her Ph.D. from Carnegie Mellon University in 2004. Her research interests are in artificial intelligence, probabilistic machine learning, and robotics.