A Tale of Two Architectures: A Dual-Citizenship Integration of Natural Language and the Cognitive Map

Tom Williams
HRI Laboratory
Tufts University
Medford, MA, USA
williams@cs.tufts.edu

Collin Johnson
Intelligent Robotics Laboratory
University of Michigan
Ann Arbor, MI, USA
collinej@umich.edu

Matthias Scheutz
HRI Laboratory
Tufts University
Medford, MA, USA
matthias.scheutz@tufts.edu

Benjamin Kuipers
Intelligent Robotics Laboratory
University of Michigan
Ann Arbor, MI, USA
kuipers@umich.edu

ABSTRACT

Vulcan and DIARC are two robot architectures with very different capabilities: Vulcan uses rich spatial representations to facilitate navigation capabilities in real-world, campus-like environments, while DIARC uses high-level cognitive representations to facilitate human-like tasking through natural language. In this work, we show how the integration of Vulcan and DIARC enables not only the capabilities of the two individual architectures, but new synergistic capabilities as well, as each architecture leverages the strengths of the other. This integration presents interesting challenges, as DIARC and Vulcan are implemented in distinct multi-agent system middlewares.

Accordingly, a second major contribution of this paper is the Vulcan-ADE Development Environment (VADE): a novel multi-agent system framework comprised of both (1) software agents belonging to a single robot architecture and implemented in a single multi-agent system middleware, and (2) "Dual-Citizen" agents that belong to both robot architectures and that use elements of both multi-agent system middlewares. As one example application, we demonstrate the implementation of the new joint architecture and novel multi-agent system framework on a robotic wheelchair, and show how this integration advances the state-of-the-art for NL-enabled wheelchairs.

Keywords

assistive technologies; robot wheelchairs; robot architectures; human-robot interaction; communication; frameworks for agents and multi-agent systems

1. INTRODUCTION

Not all robot architectures are created equal. A large number of integrated robot architectures have been developed over the past few decades, but these differ wildly in terms of the representations they use and the capabilities and behaviors they enable, which are dependent on the research objectives of their designers. This is particularly true of the Vulcan robot architecture and middleware [26] and the Distributed, Integrated, Affect, Reflection, Cognitive Robot Architecture (DIARC) [36] as implemented in the Agent Development Environment (ADE) MAS middleware. Both *Vulcan* and *DIARC* are considered fully fledged robot architectures implemented as fully fledged multi-agent systems (MAS). These architectures, however, have relatively few overlapping representations, capabilities, and behaviors. Beyond components to handle sensory data and deliver motor commands to robot bases, the architectures do not share many common components: Vulcan uses the rich spatial representations provided by the *Hybrid Spatial Semantic Hierarchy* (HSSH) to enable navigation capabilities in real-world environments, while DIARC uses high-level cognitive representations to enable human-like tasking through natural language.

In this case, however, difference begets opportunity. By integrating the *Vulcan* and *DIARC* robot architectures (through specific integration of the*Vulcan* and *ADE* MAS middlewares), we have produced a new robot architecture that is greater than the sum of its parts, with state-of-the-art navigational capabilities thanks to Vulcan, state-of-the-art linguistic capabilities thanks to DIARC, and new synergistic capabilities made possible only through this integration (e.g., navigation to locations based on complex natural language utterances with context-dependent meanings) as each architecture leverages the other's strengths. What is more, this new hybrid integrated robot architecture is implemented in the Vulcan-ADE Development Environment (*VADE*), a novel multi-MAS-system. VADE provides a useful, novel framework for multi-MAS integration through the use of so-called *Dual Citizen* agents, as we will describe.

As an example application, we have implemented our new integrated architecture on a robotic wheelchair (as shown in Fig. 1), resulting in a wheelchair that advances the state-of-the art. Intelligent wheelchairs represent an attractive application not only because they benefit from

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

Figure 1: The Vulcan Intelligent Wheelchair

what is brought to the table by both Vulcan and DIARC, but because they promise to be of great benefit to society. Within the United States alone, there are at least 3.6 million wheelchair users, 40% of whom find it difficult or impossible to control a wheelchair using a joystick [7, 12]. To make wheelchairs more accessible, many researchers are turning to Natural Language (NL) as a control modality. While such NL-enabled wheelchairs have existed for nearly forty years (e.g., [10]), even many of the most recently presented NL-enabled wheelchairs have only limited capabilities, e.g., the ability to be commanded to go forward, left, right, backwards and to stop [2, 3, 6, 19, 20, 22, 23, 31, 32, 34, 38, 40, 39, 44, 45]. There have, however, been a small number of recent wheelchairs also capable of following walls [30], entering elevators [25], traveling to nearby objects [14], or traveling to named objects and locations [13, 24, 26, 29, 33, 42].

Although the new levels of autonomy and mobility that current NL-enabled wheelchairs grant users is promising, these wheelchairs do not come close to providing the capabilities of human helpers. A human helper pushing a wheelchair can do more than travel to named locations. Human helpers learn about new locations and other entities through observation and dialogue. They have memories of events, preferences, and goals. They ask questions, make suggestions, and make conversation. Furthermore, human helpers are not troubled by environmental features like elevators, multi-floor buildings, or “the outdoors”. While NL-enabled wheelchairs will not truly rival the capabilities of human helpers anytime soon, we believe that NL-enabled wheelchairs are close to becoming *genuine helpers* that augment their users’ capabilities in order to make them effective in tasks of daily living, build rapport, and are worthy of trust. As we will show, our integration results in great progress towards this goal.

2. DIARC AND ADE

The Distributed, Integrated, Affect, Reflection, Cognitive Robot Architecture (*DIARC*) is a hybrid deliberative-reactive robotic architecture with a wide variety of high-level cognitive capabilities [36]. Of particular relevance are DIARC’s language- memory- and action-oriented components. DIARC’s language-oriented components allow robots to resolve a wide variety of referring expressions, including anaphoric and deictic expressions (e.g., “*it* is in *that* breakroom”) and referring expressions that use *descriptions* (e.g., “go to *the room across from the breakroom*”) rather than *rigid designators* that indicate their targets by name or label [46]. Furthermore, such referring expressions need not be used in the context of direct commands: interlocutors are free to use so-called *indirect speech acts* that follow conventionalized social norms (e.g.,“*I need to* go to the bathroom”), which DIARC interprets based on context [47]. DIARC’s language-oriented components leverage its memory-oriented components: some components use the POWER framework, which allows uncertain information about both known and hypothetical entities to be distributed across multiple heterogeneous knowledge bases [49, 50]; other components use a general-purpose *belief* component to perform inference on shared knowledge. This component is leveraged by DIARC’s action-oriented components, which perform high-level goal and action management capabilities.

While DIARC does have spatial reasoning and navigation components [48], these are relatively rudimentary relative to DIARC’s cognitive components. DIARC’s motion-oriented components can easily allow a robot to traverse a hallway or travel in a certain direction, but do not provide mapping capabilities, and use only rudimentary spatial representations.

DIARC is implemented in the *Agent Development Environment* (ADE) multi-agent system middleware. ADE is an architectural framework [18] that builds on previous work from multi-agent systems [5, 41] in order to support the development of individual agent architectures using distributed multi-agent system computing infrastructure. ADE treats architectural components as autonomous software agents in order to facilitate dynamic system configuration, fault tolerance and recovery, distributed computation, and autonomic computing [35, 1, 17].

3. VULCAN

The Vulcan robot architecture is a hybrid deliberative-reactive architecture that focuses on the capabilities needed for navigation in campus-like environments. These environments consist of multiple buildings with a variety of architectural styles and pedestrian conditions ranging from empty corridors to dense crowds.

The underlying map representation used in Vulcan is a hybrid metric-topological map based on the Hybrid Spatial Semantic Hierarchy (HSSH). The HSSH is a map representation that uses metric and topological representations of small-scale space – the portion of the environment within the robot’s immediate sensory horizon – to build metric and topological maps of the large-scale environment. The HSSH is a hierarchy of ontologies, where each layer in the hierarchy provides a different abstraction of space. The Local Metric layer uses a local simultaneous localization and mapping (SLAM) algorithm to build a Local Perceptual Map (LPM) of the small-scale space around the robot. The Local Topological layer parses the LPM into a set of discrete, non-overlapping *areas*. The Global Topological layer uses the *areas* detected by the Local Topological layer to build a global topological map of large-scale space. The Global Metric layer uses the global topology to construct a metric map of large-scale space.

Parsing the environment into a topological map comprised of small, discrete *areas* separates local structure from global structure, allowing the mapping problem to be factored into smaller, more easily solved parts. For example, metric SLAM only needs to be performed within the local environment because the global structure of the environment is constructed using the global topological map. Avoiding reasoning about loop closures allows metric SLAM to run in constant time because the robot's sensors have a fixed upper-bound on how much of the environment they can see.

Planning and navigation through the environment also benefit from the HSSH representation. The global topological map provides a useful representation for navigation by factoring the motion planning problem into: (a) graph search through large-scale space, and (b) metric motion planning in small-scale space. Furthermore, the sparse, symbolic representation of a topological map allows for scalable mapping of large environments [16].

Ultimately, a robotic wheelchair serves its human driver and therefore needs to reason about the human's goals and intentions. The use of a topological map is thus advantageous as it uses human-like representation of spatial knowledge, facilitating human-like spatial reasoning. But if the semantics of such a map are grounded solely in a robot's *actions* (e.g., if a map is structured solely with respect to navigation affordances), as they are in Vulcan, then a robot using it can only be commanded through reference to these actions (e.g., by specifying a series of such affordances to exploit). Vulcan currently accepts these types of commands through a point-and-click user interface. In order to allow for more natural interactions, Vulcan needs a way of grounding its representations in the types of semantics more typically seen in human *conversations* (e.g., recognizing that a certain large-scale topological location may be a "kitchen", 'may be 'large", and may contain various goal-relevant objects), and should be able to accept commands that reference those aspects through a *natural language* interface.

Vulcan is implemented as a set of asynchronous, distributed components which communicate through a publish-subscribe model, using the LCM communication library [15]. Though this approach lacks several features typical to multi-agent systems, including white-page and yellow-page functionality, we argue the Vulcan Middleware is a MAS when viewed within the context of our larger integrated architecture.

4. INTEGRATED APPROACH

4.1 The Vulcan-ADE Development Environment (VADE)

The Vulcan-ADE Development Environment (VADE) framework is a multi-agent system comprised of three types of components:

- **DIARC Components:** ADE components that only exist within the DIARC architecture, and are only aware of components implemented in the ADE middleware.
- **Vulcan Components:** Vulcan components that only exist within the Vulcan architecture, and are only aware of components communicating on Vulcan's LCM channels.
- **New Dual-Citizen Components:** Components that exist within *both* architectures, and can communicate both with ADE components through Java RMI and with Vulcan Components through LCM.

VADE's Dual-Citizen Components are functional components that require information and/or capabilities from both DIARC and Vulcan in order to provide their desired functions. These components exist within both architectures, and are aware of both multi-agent systems, thus effecting an inter-architectural bridge while maintaining flexibility and preventing single-architecture components of either architecture from needing to know about the single-architecture components of the other architecture.

These components are implemented as Java classes that both extend the ADE Component interface (allowing communication with ADE Components and the ADE Registry) and provide LCM Publisher/Subscriber interfaces (allowing communication with Vulcan Components). Of course, these components cannot *physically* extend to both architectures; we thus choose to grant them "primary" citizenship within the ADE multi-agent system framework, so that they can be started by the ADE Registry. This also means that if these components fail, they can be restarted automatically by the ADE Registry. If this happens, they will automatically resubscribe to the appropriate LCM channels, allowing Publishing and Subscribing functionality to automatically go back into effect.

Of course, this is not the only choice we could have made. One (expensive) option would have been to reimplement all of Vulcan architecture within the ADE multi-agent system middleware or all of DIARC architecture within the Vulcan middleware. However, this would not only have been monumentally time consuming, but would have removed functionality. Reimplementing Vulcan within ADE would have eliminated the speed advantages crucial to Vulcan's operations at the hardware level; reimplementing DIARC within Vulcan would have removed the OS-agnostic portability, easy distributability, and middleware features (e.g., dynamic system reconfiguration) afforded by ADE.

Another option would have been to implement a "bridge" component that handles all inter-architecture traffic. However, this would have been problematic for two reasons: (1) it would have created a computational bottleneck, and (2) in the case of failure of this component, all inter-architectural communication would necessarily cease. In contrast, if one Dual-Citizen Component goes down, other inter-architectural communication can still proceed as usual while the failed component goes through the process of restarting and reconnecting with both architectures.

4.2 Vulcan-DIARC implementation in VADE

There are a number of advantages to Vulcan-DIARC integration at the architectural level: by integrating Vulcan and DIARC, each can leverage the other's capabilities, resulting in new synergistic capabilities and behaviors. DIARC alone is unable to engage in dialogue regarding large-scale spatial locations, not because it lacks the linguistic faculties, but rather because it lacks significantly rich spatial representations – such representations can be provided by Vulcan.

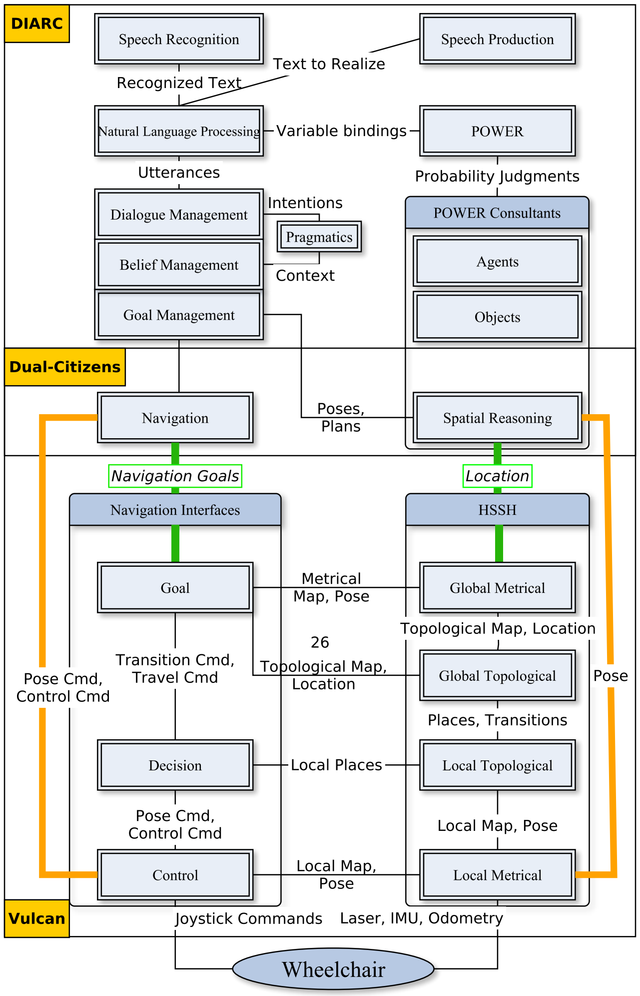

Figure 2: Diagram for the integrated system. The top and bottom halves correspond with the DIARC and Vulcan architectures, respectively, with the Navigation and Spatial Reasoning components in the overlapping Dual-Citizen region belonging to both architectures. Orange lines represent the existing inter-architecture connections. Green lines represent intended future connections. Connections between the Dialogue, Belief, and Goal Managers are not shown due to connection density.

Similarly, we have typically restricted DIARC to small, simple, indoor environments – DIARC can leverage Vulcan's spatial reasoning and mapping capabilities in order to discuss, reason about, and travel through larger environments.

Similarly, Vulcan requires commands to be precisely specified within its map representation, e.g., using metric coordinates, a topological action, or a named topological location. But by leveraging DIARC's linguistic capabilities, Vulcan can travel to locations that are only loosely specified; an NL comment like "I need my coffee!" does not clearly specify a location, yet can be used to both infer such a location, and the fact that the robot should travel to it.

As shown in Figure 2, the majority of VADE components are pure Vulcan and ADE components. with two components serving as Dual-Citizen components: the Navigation Component and the Spatial Reasoning Consultant Component. On the ADE side, these interact, respectively, with the Goal Management and POWER components, as we will describe. On the Vulcan side, we would ultimately like to effect integration at multiple levels of the HSSH so that DIARC can leverage Vulcan's rich topological representations. At this point, however, these dual citizen components only communicate, respectively, with Vulcan's Control and Local Metrical components. This provides DIARC with access to pose representations maintained by Vulcan, which can be used to determine the robot's current topological location within DIARC's own internal topological map (maintained in the Spatial Reasoning Consultant ADE Component). DIARC is then able to navigate through its internal topological map by sending motion target to the Vulcan motion planner when it decides it needs to visit the a particular topological location. A result of this integration is that DIARC is now able to operate in more dynamic and difficult environments.

In the following sections, we will go into depth into VADE's Components. As we have previously stated, we use a robot wheelchair as one example application of our integrated approach. Before discussing the software modules, we first discuss the specific hardware architecture of the wheelchair.

4.3 Hardware Architecture

The presented robotic wheelchair (Fig. 1) is a commercially-available powered wheelchair (Quantum 6000Z) modified to enable computer control and augmented with two Hokuyo UTM-30LX laser rangefinders (one at the front-right corner and one at the back-left corner) to provide a 360° view of the wheelchair's surroundings. Wheel encoders and an inertial measurement unit are mounted on the wheelchair to enable high-precision motor control.

The wheelchair is driven using a joystick. We enable computer control by intercepting CAN bus communication between the joystick and on-board controller. During autonomous driving, the commands sent by the joystick are replaced by commands calculated by Vulcan's motion controller. Whenever the joystick is in use by the driver, the command from the human-controlled joystick is used. Otherwise, the command from the Vulcan motion controller is used. We thus always defer to human control.

4.4 Vulcan Components

The Vulcan robot architecture separates the overall problem of mapping, localizing, and navigating into two modules, as shown in the lower half of Fig. 2.

4.4.1 HSSH

The robot's map is represented using a variation of the Hybrid Spatial Semantic Hierarchy (HSSH) [4]. The HSSH is a hybrid metric-topological map representation containing a hierarchy of spatial ontologies that provide metric and topological representations of *small-scale* and *large-scale* space.

The *Local Metric* layer represents the portion of the environment within the robot's sensory horizon using an occupancy-grid-based representation called the Local Perceptual Map (LPM). The LPM contains the robot's metric knowledge of its current topological area. When the robot transitions between topological areas, the LPM is shrunk to remove the previous area. Discarding previously visited areas ensures that no large-scale loop closures occur within the LPM.

The *Local Topological* layer detects and classifies places within the LPM. Through this process, a symbolic description of the LPM is created that contains a set of discrete, non-overlapping *areas*: the basic entities in Vulcan's topological map representation. This representation extends the HSSH's in the following ways.

First, the Vulcan representation distinguishes between two types of places: decision points and destinations. A *decision point* is an area located at the intersection of two or more paths. A *destination* is an area that corresponds to a location like an office or conference room. Decision points are essential for determining the overall connectivity of areas in the environment and are essential for navigation. Destinations are most often located along paths and are usually visited only if the robot is commanded to travel to them.

The second extension to the HSSH is a richer representation for *path segments*. Before, a path segment was a simple connector that provided travel between two places. Now, one can also have destinations located along its sides. This allows for a more natural representation of the common indoor environments encountered by the wheelchair.

The *Global Topological* layer creates and maintains a globally consistent topological map of the areas in the environment by running a topological SLAM algorithm [16] as the robot travels through the environment. When the robot visits an area, the Global Topological layer is responsible for finding possible matches between the new area and a previously visited one, and adding new areas to the map.

4.4.2 Navigation Interfaces

The *Control* interface interacts with the Local Metric layer of the HSSH, allowing the user to specify a velocity, (v, ω), for the robot to drive at, or a specific target pose, (x, y, θ), to drive to. In either case, the robot avoids obstacles and safely executes the provided command.

For Vulcan, the Control interface is implemented using a model-predictive control algorithm, MPEPC [28, 27], capable of navigating through the inherently dynamic and uncertain conditions present in the real world. MPEPC generates plans by simulating a variety of possible actions for the robot to take and choosing a locally optimal action. The cost function used to make this choice includes the cost of the robot's actions and the probability of collision with static and dynamic obstacles, taking into account uncertainty about both the robot's motion and the motion of pedestrians.

The *Decision* interface interacts with the Local Topological layer of the HSSH and allows a user to choose amongst a small selection of qualitative actions for the robot to take within a local area. For example, the robot can be commanded to drive to the end of a path segment and then turn left when it reaches the next decision point. In this way, the Decision interface allows the robot to be commanded in a similar fashion to how a person might give directions to another person [21]: by specifying a sequence of actions to take at decision points along a route. The commands are specified using the keyboard or a graphical interface.

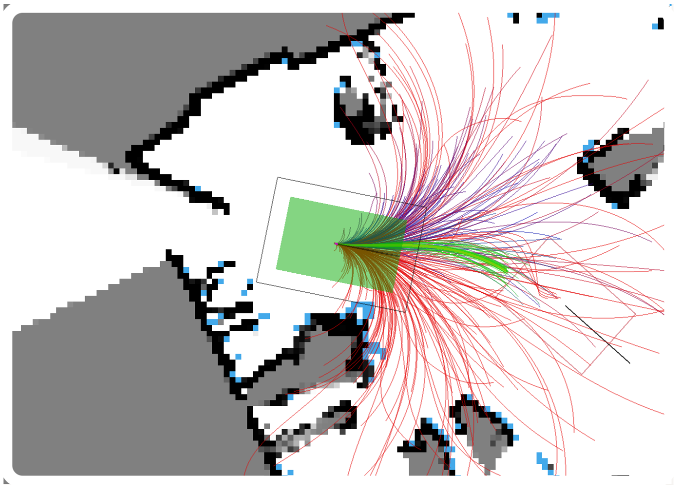

Figure 3: Visualization of Vulcan's metric path planning during execution of a navigation command.

The *Goal* interface interacts with the Global Topological and Global Metric layers of the HSSH. The Goal interface allows user to command the robot to drive to some named location. In the global topological map, this location will be a path or place that has been labeled with a unique name, like a bathroom or office. In the global metric map, this location would be a pose, (x, y, θ), within a place like a bathroom or office. Goals are selected using a graphical interface to select amongst the collection of previously named locations.

4.5 ADE Components

In this subsection we will describe VADE's ADE components, as shown in the upper half of Fig. 2.

4.5.1 Speech Recognition and Production

Natural language utterances enter the architecture through DIARC's *Speech Recognition* component, which uses the Sphinx4 library [43] to transduce speech into text. Similarly, the *Speech Production* component uses the MaryTTS library [37] to synthesize text into vocal output.

4.5.2 Natural Language Processing (NLP)

The NLP component first performs syntactic processing using the C&C CCG-based dependency parser [11]. The produced dependency graph is then converted to a tree which is used for several purposes [46]: (1) One variable is instantiated for each referenced entity in the tree; (2) Logical formulae denoting properties and relations are instantiated for each property and relation in the tree, and semantic processing rules are used to analyze the tree in order to produce a formula for the tree's root node; (3) "Status cues" are associated with each referenced entity, based on what determiner (if any) is attached to that entity; (4) The utterance's illocutionary point (e.g., Statement, Question, Instruction) is determined based on the root node of the tree.

4.5.3 Reference Resolution

Reference resolution determines what entities in the robot's (possibly distributed and heterogeneous [50]) knowledge bases should be associated with each *referenced* entity, using the Probabilistic Open World Entity Resolution (POWER) algorithm [49]. POWER is designed to operate in uncertain and open worlds, and handles references to both known and unknown entities. Specifically, the Givenness-Hierarchy-theoretic *GH-POWER* is used [46], which uses a hierarchical cognitively-inspired memory structure (consisting of the *Focus of Attention*, *Short Term Memory*, *Discourse Context*, and *Long Term Memory*) to resolve definite noun phrases and anaphoric and deictic expressions.

The end product of reference resolution is a set of mappings from variables to memory traces associated with entities in a robot's long-term memory. These memory traces are used to create a set of bound semantic structures, which differentially bind the open variables of the logical formula associated with the utterance's root node. These bound semantic structures are used in turn to create bound utterance representations which are sent to the pragmatics component.

4.5.4 Pragmatics

DIARC's Pragmatics component uses a set of context-sensitive Dempster-Shafer-Theoretic logical rules to determine the intention underlying each candidate utterance representation [47]. This results in a set of *belief updates* which are passed to the Dialogue, Belief, and Goal Management components (DBGM, collectively).

4.5.5 Dialogue, Belief, and Goal Management

DIARC's Dialogue, Belief, and Goal Management components (DBGM, collectively) are responsible for tracking and coordinating dialogue [9], storing beliefs and performing inference in a general-purpose knowledge base, and tracking and acting on goals [8]. If the DBGM needs to respond to its user, it sends its own intention back through the Pragmatics component, which can work in reverse to determine the utterance which should be used to communicate a particular intention. If a robot's interlocutor uses a command to instruct the robot, the DBGM instantiates a new goal based on intentions underlying that command and determines how to accomplish it. Of particular relevance are commands to travel to particular locations, which is accomplished by DIARC's Spatial Reasoning Component.

4.6 Dual-Citizen Components

The Spatial Reasoning Consultant Component (SRC) serves as the primary *knowledge* interface between DIARC and Vulcan. The SRC maintains a graph of both large-scale and small-scale topological locations, where connectivity indicates either physical adjacency or containment. Each location is associated with an identifier, property list, and (in the case of grounded small-scale locations), coordinate pose.

When DIARC determines that the robot must navigate to a particular location, the SRC finds the shortest path to that location, and incrementally sends the robot to the coordinates of each intermediate waypoint by sending them to the Navigation Component, which serves as the primary *action* interface between DIARC and Vulcan. The Navigation Component broadcasts those coordinates over the appropriate LCM channel. Similarly, when DIARC's goal manager determines the robot needs to simply drive forward, turn, or stop, the navigation component effects these motions by broadcasting messages over other LCM channels. The next integration step will be to integrate Vulcan's topological capabilities with DIARC: when this is accomplished, many of the responsibilities of these two components will be transferred to pure Vulcan components, providing more of a communicational role to these two Dual-Citizen Components.

Note that both DIARC and the HSSH are fully fledged robot architectures, and capabilities implemented in their respective middlewares. For example, both architectures typically make use of the robot's laser rangefinders, for example, and both typically send motor commands to effect robot motion. But both architectures cannot be responsible for these overlapping capabilities, and thus we had to decide which architecture should cede some of its control. Because the Vulcan solely focuses on spatial reasoning and motion planning, it makes sense for DIARC to do so. From DIARC's perspective, each Dual-Citizen component is just another component, which happens to provide these ceded capabilities, and is not aware that those components are in fact part of an entire another architecture, or that the motion primitives sent to the Navigation component may spawn complex navigational procedures. Similarly, Vulcan views each Dual-Citizen components as just another publisher/subscriber, and is not aware that the motion targets they publish come from DIARC and not a human user.

This is not to say that there are not disadvantages to this approach, however. Because Vulcan and ADE middlewares are not aware that they are part of the larger VADE middleware, VADE essentially has two managers that are not aware of each other. Each manager may take actions that are locally beneficial but deeply problematic from the perspective of the other manager. If the ADE registry decides to perform dynamic load balancing by moving a Dual-Citizen component to a different host, this would be equivalent to that component going down from the perspective of Vulcan. While these problems are ameliorated within VADE by the decentralized nature of Vulcan, such cases may still occur if Dual-Citizen components are started on or dynamically relocated to a host to which Vulcan does not have access permissions. As a result, it may be necessary to proactively attempt to avoid some of these failure cases through changes in architectural configuration, or disabling of certain middleware capabilities – considerations that will need to be more seriously considered by MAS integrators intending to use a Dual-Citizen approach with a different pair of MAS middlewares.

5. DEMONSTRATION

We will now present a proof of concept demonstration of our integrated approach, as implemented on the robotic wheelchair we previously described. A video of this demonstration can be viewed at `https://tiny.cc/wheelchairdemo`. In this demonstration, the wheelchair begins in an office environment, and is told by its rider ("Jim") "I need my coffee!". After recognition, DIARC's ASR component passes this utterance to its NLP component, which performs parsing and reference resolution. This utterance is parsed into the utterance form

$Statement(jim, self, need(jim, X))$ with supplemental semantics $coffee(X)$.

At the start of this interaction, the robot's *Short Term Memory* and *Focus of Attention* are both empty, and thus the robot's *Long Term Memory* is searched for a suitable referent to bind to the variable X. The property $coffee(X)$ is advertised only by POWER's *objects* consultant, which manages a knowledge base of known objects. This KB starts off with knowledge of a handful of objects, including their properties and locations. Included in this set is one coffee-like entity, with memory trace obj_5. This trace is bound to X, producing $Statement(jim, self, need(jim, obj_5))$, which is passed to DIARC's pragmatic reasoning component. This component has a rule with implicative content:
$Statement(X, Y, need(X, Z)) \Rightarrow goal(Y, have(X, Z))$, resulting in the goal $have(jim, obj_5)$ being adopted.

The SRC identifies the location of obj_5 as loc_{51}, and creates a plan to visit the set of waypoints on the path to loc_{51}. DIARC's DBGM then executes this plan one step at a time: for each waypoint, the DBGM acquires the coordinates of that location from the SRC and passes them to DIARC's Navigation component, which in turn broadcasts these coordinates over LCM to Vulcan.

The command from the Navigation component is received by Vulcan's Control component, which initiates a new motion planning task to drive to the specified coordinates. Using MPEPC, the wheelchair performs the task by driving to the desired coordinates. During motion planning, the state of the environment is estimated at 10Hz, including the position and velocity of pedestrians around the robot, the location of static obstacles, and the wheelchair's own position and velocity. This fast update allows the wheelchair to safely navigate even through dense crowds.

When the wheelchair arrives at a destination, it broadcasts an LCM message indicating action success. When this message is received by the Dual-Citizen Navigation component, it moves on to the next step in its navigation plan: once again, the DBGM will acquire the coordinates of the next small-scale place along the path, and send those coordinates to the Navigation component. As this process iterates, the wheelchair drives down several hallways until it reaches the door to the room containing the coffee. The robot then turns, and drives through the doorway in order to reach the last point along the route.

We have demonstrated the capabilities enabled by our integrated approach: this should not be taken as, nor is this intended to be, a formal empirical evaluation. While the components of the Vulcan and DIARC architectures have been evaluated independently, a holistic, extrinsic evaluation of this integrated approach will still eventually be necessary.

6. CONCLUSIONS

In this paper, we make three primary contributions. First, we demonstrated how the integration of the Vulcan and DIARC architectures produces a hybrid architecture with not only the capabilities of both architectures, but new synergistic capabilities as well: by leveraging Vulcan, DIARC can navigate through environments, Vulcan can initiate actions based on flexible natural language requests, and as a whole, Vulcan-DIARC can now travel to previously unknown objects that are learned about in one-shot through natural language, a capability previously held by neither architecture.

Figure 4: Metric-topological map of the demonstration environment. This 44m x 75m map was produced by running Vulcan's metric SLAM and place classification algorithms on sensor data of the wheelchair being manually driven through an environment. Color indicates the "type" of each region in the topological map: decision points are blue, path segments are green, and destinations are red.

Second, we showed how this integration could be implemented in a new multi-agent system comprised of agents from two distinct multi-agent systems, plus Dual-Citizen agents that belonged to *both* multi-agent systems. This provides a novel, useful framework for multi-MAS integration which could be used for future integrated approaches.

Finally, we showed how, when implemented on a robotic wheelchair, this integration significantly extends the state-of-the-art for NL-enabled wheelchairs. Like a small number of other recent wheelchairs [13, 24, 26, 29, 33, 42], our wheelchair can travel to described objects and locations. Within this set of wheelchairs, however, our is unique with respect to its cognitive approach: to our knowledge, no previously presented NL-enabled wheelchair has been capable of handling natural, indirect language [47], hypothesizing new objects and locations based on natural language [50], modeling cognitive structures to resolve anaphora [46], or asking clarification questions [47, 51], all of which are afforded to our wheelchair through this integrated approach.

As previously discussed, however, there are a number of architectural interfaces that have not yet been implemented, most notably the use of the HSSH's rich topological representations within DIARC, and the integration of Vulcan's Decision and Goal interfaces. This represents our immediate next step for future work. Integrating these remaining architectural interfaces will in turn enable a host of possible research directions. Allowing DIARC to add new abstract representations for large-scale locations to Vulcan's topological map will allow a DIARC-Vulcan controlled robot to travel to not only previously unknown *objects* described in natural language, but to previously unknown *locations* as well (c.f. [48]). In the future, we are also interested in integrating novel episodic memory management and preference modeling capabilities into DIARC. This would allow a DIARC-Vulcan controlled robot to follow directives such as "Bring me to *my barbershop*" or "Let's go to *the park we visited last week*," which require consideration of such episodic memories or preferences. Once such capabilities have been enabled, we aim to perform a long-term, extrinsic usability study evaluating the wheelchair's ability to facilitate the tasks of everyday living. Of course, the presented application to a robotic wheelchair is only one example; the integration of these two architectures may well lead to significant advances in other domains as well.

ACKNOWLEDGMENTS

Research at the University of Michigan Intelligent Robotics Laboratory is supported in part by grants from the National Science Foundation (IIS-1111494 and IIS-1421168); Research at the Tufts University Human-Robot Interaction Laboratory is funded in part by grants from the National Science Foundation (IIS-1111323) and the United States Office of Naval Research (#N00014-11-1-0493 and #N00014-14-1-0149). In addition, the authors would like to thank Daniel de Cordoba Gil, Jong Jin Park, Monica Dragulski, Ronald Gaynier, and Zachary Haga for their assistance over the course of this project.

REFERENCES

[1] V. Andronache and M. Scheutz. ADE — an architecture development environment for virtual and robotic agents. *International Journal on Artificial Intelligence Tools*, 15(02):251–285, 2006.

[2] T. Asakawa and K. Nishihara. Operation assistance of a voice-controlled electric wheelchair. In *Proceedings of ICMIT 2007*, 2007.

[3] O. Babri, S. Malik, T. Ibrahim, and Z. Ahmed. Voice controlled motorized wheelchair with real time obstacle avoidance. In *ICCIT 2012*, 2012.

[4] P. Beeson, J. Modayil, and B. Kuipers. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *The International Journal of Robotics Research*, 29(4):428–459, 2010.

[5] F. Bellifemine, A. Poggi, and G. Rimassa. Jade–a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.

[6] R. Berjon, M. Mateos, A. Barriuso, I. Muriel, and G. Villarrubia. Alternative human-machine interface system for powered wheelchairs. In *Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on*, pages 1–5. IEEE, 2011.

[7] M. W. Brault. Americans with disabilities: 2010. *Current Population Reports*, 7:0–131, 2012.

[8] T. Brick, P. Schermerhorn, and M. Scheutz. Speech and action: Integration of action and language for mobile robots. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1423–1428, San Diego, CA, October/November 2007.

[9] G. Briggs and M. Scheutz. Multi-modal belief updates in multi-robot human-robot dialogue interaction. In *Proceedings of 2012 Symposium on Linguistic and Cognitive Approaches to Dialogue Agents*, 2012.

[10] J. A. Clark and R. B. Roemer. Voice controlled wheelchair. *Archives of Physical Medicine and Rehabilitation*, 58(4):169–175, 1977.

[11] S. Clark and J. R. Curran. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33(4):493–552, 2007.

[12] L. Fehr, W. E. Langbein, and S. B. Skaar. Adequacy of power wheelchair control i terfaces for persons with severe disabilities: A clinical survey. *Development*, 37(3):353–360, 2000.

[13] S. Hemachandra, T. Kollar, N. Roy, and S. Teller. Following and interpreting narrated guided tours. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[14] B. A. Hockey and D. P. Miller. A demonstration of a converstionally guided smart wheelchair. In *Proceedings of ASSETS '07*, pages 243–244, 2007.

[15] A. S. Huang, E. Olson, and D. C. Moore. Lcm: Lightweight communications and marshalling. In *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*, pages 4057–4062. IEEE, 2010.

[16] C. Johnson and B. Kuipers. Efficient search for correct and useful topological maps. In *IROS*, pages 5277–5282. IEEE, 2012.

[17] J. Kramer and M. Scheutz. Ade: A framework for robust complex robotic architectures. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4576–4581. IEEE, 2006.

[18] J. Kramer and M. Scheutz. Development environments for autonomous mobile robots: A survey. *Autonomous Robots*, 22(2):101–132, 2007.

[19] L. H. Linh, N. T. Hai, N. Van Thuyen, T. T. Mai, and V. Van Toi. MFCC-DTW Algorithm for Speech Recognition in an Intelligent Wheelchair. In *5th International Conference on Biomedical Engineering in Vietnam*, pages 417–421, Cham, 2015. Springer International Publishing.

[20] J. Liu, H. Zhang, B. Fan, G. Wang, and J. Wu. A novel economical embedded multi-mode intelligent control system for powered wheelchair. In *Computing, Control and Industrial Engineering (CCIE), 2010 International Conference on*, volume 1, pages 156–159. IEEE, 2010.

[21] M. MacMahon, B. Stankiewicz, and B. Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI'06*, pages 1475–1482. AAAI Press, 2006.

[22] R. Maskeliunas and R. Simutis. Multimodal wheelchair control for the paralyzed people. *Elektronika ir Elektrotechnika*, pages 81–84, 2011.

[23] C. McMurrough, I. Ranatunga, A. Papangelis, D. O. Popa, and F. Makedon. A development and evaluation platform for non-tactile power wheelchair controls. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, page 4. ACM, 2013.

[24] R. Megalingam, R. Nair, and S. Prakhya. Automated voice based home navigation system for the elderly and the physically challenged. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1 –5, 28 2011-march 3 2011.

[25] A. Murai, M. Mizuguchi, T. Saitoh, T. Osaki, and R. Konishi. Elevator available voice activated wheelchair. In *Robot and Human Interactive*

Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on, pages 730 –735, 27 2009-oct. 2 2009.

[26] A. Murarka, S. Gulati, P. Beeson, and B. Kuipers. Towards a safe, low-cost, intelligent wheelchair. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV)*, pages 42–50. Citeseer, 2009.

[27] J. J. Park. *Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments*. PhD thesis, University of Michigan – Ann Arbor, 2016.

[28] J. J. Park, C. Johnson, and B. Kuipers. Robot navigation with model predictive equilibrium point control. In *IROS*, pages 4945–4952, 2012.

[29] M. R. Petry, A. P. Moreira, B. M. Faria, and L. P. Reis. Intellwheels: intelligent wheelchair with user-centered design. In *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, pages 414–418. IEEE, 2013.

[30] S. Png and J. Pineau. Bayesian reinforcement learning for pomdp-based dialogue systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2156 –2159, 2011.

[31] M. Qadri and S. Ahmed. Voice controlled wheelchair using dsk tms320c6711. In *Signal Acquisition and Processing, 2009. ICSAP 2009. International Conference on*, pages 217–220. IEEE, 2009.

[32] U. Qidwai and F. Ibrahim. Arabic speech-controlled wheelchair: A fuzzy scenario. In *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pages 153 –156, 2010.

[33] T. Röfer, C. Mandel, A. Lankenau, B. Gersdorf, and U. Frese. 15 years of rolland. In *Festschrift Dedicated to Bernd Krieg-Brückner on the Occasion of his 60th Birthday*, pages 255–272, 2009.

[34] A. Ruíz-Serrano, R. Posada-Gómez, A. M. Sibaja, G. A. Rodríguez, B. Gonzalez-Sanchez, and O. Sandoval-Gonzalez. Development of a dual control system applied to a smart wheelchair, using magnetic and speech control. *Procedia Technology*, 7:158–165, 2013.

[35] M. Scheutz. ADE - steps towards a distributed development and runtime environment for complex robotic agent architectures. *Applied Artificial Intelligence*, 20(4-5):275–304, 2006.

[36] M. Scheutz, G. Briggs, R. Cantrell, E. Krause, T. Williams, and R. Veale. Novel mechanisms for natural human-robot interactions in the diarc architecture. In *Proceedings of AAAI Workshop on Intelligent Robotic Systems*, page forthcoming, 2013.

[37] M. Schröder and J. Trouvain. The german text-to-speech synthesis system mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377, 2003.

[38] S. A. M. S. Sheikh and D. R. Rotake. An evolutionary approach for smart wheelchair system. In *Communications and Signal Processing (ICCSP), 2015 International Conference on*, pages 1811–1815. IEEE, 2015.

[39] A. Škraba, R. Stojanović, A. Zupan, A. Koložvari, and D. Kofjač. Speech-controlled cloud-based wheelchair platform for disabled persons. *Microprocessors and Microsystems*, 39(8):819–828, 2015.

[40] S.-Y. Suk, H.-Y. Chung, and H. Kojima. Voice/non-voice classification using reliable fundamental frequency estimator for voice activated powered wheelchair control. In Y.-H. Lee, H.-N. Kim, J. Kim, Y. Park, L. Yang, and S. Kim, editors, *Embedded Software and Systems*, volume 4523 of *Lecture Notes in Computer Science*, pages 347–357. Springer Berlin / Heidelberg, 2007.

[41] K. Sycara, M. Paolucci, M. Van Velsen, and J. Giampapa. The retsina mas infrastructure. *Autonomous agents and multi-agent systems*, 7(1-2):29–48, 2003.

[42] Y. Tao, T. Wang, H. Wei, and D. Chen. A behavior control method based on hierarchical pomdp for intelligent wheelchair. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 893–898. IEEE, 2009.

[43] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.

[44] F. Wallam and M. Asif. Dynamic finger movement tracking and voice commands based smart wheelchair. *International Journal of Computer and Electrical Engineering*, 3(4):497, 2011.

[45] H. Wang, T. Li, and F. Zheng. A Wheelchair Platform Controlled by A Multimodal Interface. In *2nd International Conference on Information Science and Control Engineering*, pages 587–590, 2015.

[46] T. Williams, S. Acharya, S. Schreitter, and M. Scheutz. Situated open world reference resolution for human-robot dialogue. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction*, 2016.

[47] T. Williams, G. Briggs, B. Oosterveld, and M. Scheutz. Going beyond literal command-based instructions: Extending robotic natural language interaction capabilities. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[48] T. Williams, R. Cantrell, G. Briggs, P. Schermerhorn, and M. Scheutz. Grounding natural language references to unvisited and hypothetical locations. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, July 2013.

[49] T. Williams and M. Scheutz. Power: A domain-independent algorithm for probabilistic, open-world entity resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[50] T. Williams and M. Scheutz. A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.

[51] T. Williams and M. Scheutz. Resolution of referential ambiguity using dempster-shafer theoretic pragmatics. In *AAAI Fall Symposium on AI and HRI*, 2016.