Bridging the Gap Between Semantic Planning and Continuous Control for Mobile Manipulation Using a Graph-Based World Representation

Roland Philippsen[*†], **Negin Nejati**[‡], **Luis Sentis**[†]
[†]Stanford Robotics and AI Lab, [‡]Stanford Computational Learning Lab
roland.philippsen@gmx.net, negin@stanford.edu, lsentis@stanford.edu

Abstract

We present our ongoing efforts to create a mobile manipulation database tool, a flexible multi-modal representation supporting persistent life-long adaptation for autonomous service robots in every-day environments. Its application to a prototypical domain illustrates how it provides symbol grounding to a reasoning system capable of learning new concepts, couples semantic planning with whole-body prioritized control, and supports exploration of uncertain and dynamic environments.

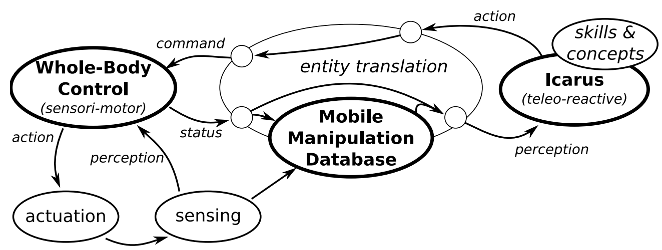

Figure 1: Overview of the system components implemented for the proof of concept presented in this paper, with arrows denoting information flow.

1 Introduction

In this position paper, we describe our approach for bridging the gap between a sensori-motor control framework, developed at the Stanford Robotics Lab, and a flexible symbolic teleo-reactive reasoning system, developed at the Stanford Computational Learning Lab. The context of our research is to integrate several areas of research in autonomous systems, employing learning, planning, perception, and control, to achieve task-oriented whole-body motions for a robot's physical interaction in environments shared with humans.

Our position is that the scalability of skills required for autonomous mobile manipulation in everyday environments can be achieved by combining a reasoning and learning system built on goal-indexed hierarchical task networks with a continuous control framework capable of handling physical interaction behaviors, and that this combination requires a representation that is rich enough to encompass information relevant to both of these components yet lightweight enough to act as a "live" model of the world and the robot.

We rely on a smart database that serves as a white-board between components, in order to *(i)* ground symbols in the robot's low-level sensing and action capabilities, *(ii)* maintain and share information relevant to more than one component, and in the long run *(iii)* support life-long adaptability of the robot in changing and uncertain environments.

Figure 1 illustrates the system architecture. The *mobile manipulation database* (MMDB) is the central component that allows *entity translation* to be a very lightweight process. *Whole-body control* (WBC) implements local continuous-domain task-oriented behaviors, while *Icarus* provides symbolic commands based on global goals and the current state of the world. This architecture is a draft that is strongly inspired by the classical three-tiered approach. It serves the purpose of illustrating a proof of concept, but the boundaries between layers are not strictly defined: Icarus is a teleo-reactive system that encompasses reasoning and execution, and the MMDB handles information that pertains to all components of the system.

Robotics and AI researchers have investigated everyday manipulation tasks for a long time, and addressing this application requires integrating approaches from several subfields. Thus, the amount of related work is quite vast, and we limit ourselves to an overview of the contributions that have most strongly influenced our collaboration so far. Related work is given at the beginnings of the sections presenting Icarus, WBC, and MMDB.

Part of the integration challenge stems from differing technical terms and unstated assumptions. Thus, one of the starting points is to define common terminology, of which we give a very short summary here.

Tasks can mean either *(i)* high-level specifications of desired actions expressed in formal logic clauses (in Icarus), or *(ii)* a task is a specific facet of continuous control (in WBC). A **goal** is a set of states which correspond to a desired outcome, where the achievement of the goal must be measurable. **Planning** is the process of finding a sequence of actions to take the system from a known initial state to a spec-

[*]This work has been supported by the Swiss National Science Foundation under the Fellowship for Advanced Researchers, grant number 115346

Table 1: Examples of percepts, concepts and skills in Icarus.

Percepts

```
(object 26 tag window state clean)
(close-to 29)
```

A concept definition example

```
((all-windows-clean)
  :percept ((object ?window tag window))
  :relations ((not (dirty ?window))))
```

A skill definition example

```
((clean ?window)
  :subgoals ((close-to ?window)
    (action-clean ?window)))
```

ified goal state. **Control** refers to real-time computation of actuator commands such that operational constraints are satisfied. **Manipulation** refers to actions that *(i)* influence the arrangement of objects in the robot's environment or *(ii)* interact physically with other agents (e.g. helping a person cross the street). **Perception** acquires and interprets information about the environment, yielding information grounded in sensor readings and ranging from low-level geometrical features to high-level abstract states.

The whole-body controller is not built on formal logic, and thus it is necessary to define a taxonomy of behaviors it can execute in order to integrate it with Icarus. This is an ongoing process, which gives each whole-body behavior a *name*, lists what *task types* it includes (e.g. table 2), specifies the types and ranges of *parameters* it accepts (e.g. controller gains, velocity and acceleration bounds), what it's *inputs* are (e.g. desired posture, visual feedback), and what kind of *output* or feedback it provides to higher levels.

2 Icarus: Reactive Symbolic Planning

An autonomous agent needs the ability to robustly choose actions that lead to its goal while continuously considering its changing perceptions of the dynamic environment. Teleo-reactive programming is a formalism for computing and organizing actions for an agent that provides this capability [Nilsson, 1994]. Icarus [Langley and Choi, 2006] is a cognitive architecture for physical agents with a commitment to teleo-reactive logic programs as the representation which supports execution and acquisition of complex procedures (figure 2). At each cycle, Icarus perceives the environment, recognizes situations, and chooses an action based on the situation and the goal, aided by two knowledge bases: *(i)* **conceptual** knowledge allows recognizing relevant situations and describes them in a higher level of abstraction, and *(ii)* **skill** knowledge encodes how the agent can affect its environment. Concepts are encoded as hierarchical monotonic inference rules with a syntax similar to Horn clauses. Skills are represented with **goal-indexed** Hierarchical Task Networks (HTNs) [Nau *et al.*, 2003]. Each skill is a recipe for decomposing a high level task into lower level ones, providing a partial ordering between them, and specifying a precondition that needs to be satisfied in the environment before it can get selected. We chose goal-indexed HTNs as skill representation because *(i)* they provide transferable solutions to similar

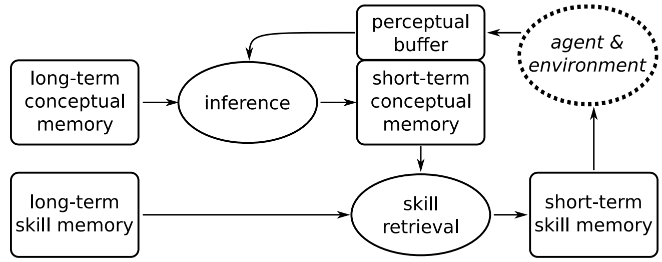

Figure 2: The Icarus cognitive architecture employs long- and short-term memories for concepts and skills in order to produce goal-directed actions that take into account a changing and uncertain environment.

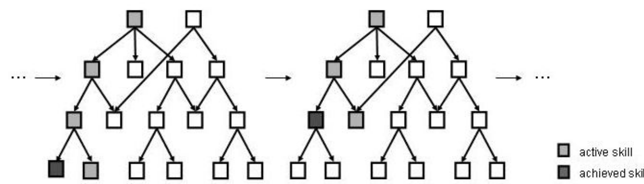

Figure 3: Icarus uses HTNs for teleo-reactive skill-selection. Skills are goal-indexed: at the highest level are the global goals, and primitive skills are leave nodes corresponding to executable actions.

problems and *(ii)* they can be automatically acquired. Table 1 provides examples for percepts, concepts and skills.

Once a goal is chosen, Icarus works out the relationship among different actions and propose a suitable one at each step (see figure 3). These hierarchies are built by Icarus dynamically at each runtime cycle and the skill path is selected in a top-down manner starting at the skill indexed by the intended goal and rooted in a primitive skill which is applicable in the current state of the world. This is more scalable than traditional controller programming: once new behaviors and goals are added into the system, Icarus can automatically use them together with the previous knowledge. Icarus keeps the skill selection path as similar as possible across cycles, but if a previously achieved goal becomes untrue again, it can interrupt its current activity to re-achieve the older goal.

Goal-indexed HTNs can be time consuming to craft, making it worthwhile to investigate automatic ways of acquiring them [Choi and Langley, 2005], [Nejati *et al.*, 2006]. The latter introduces LIGHT, an approach for HTN learning by observing sequences of operators taken from expert solutions to a problem. By analyzing the solution in the context of a background knowledge, LIGHT learns skills for achieving complex tasks, their preconditions, and partial ordering among their subgoals. It is important to note that the hierarchical nature of the learned skills is crucial for scalability as shown in [Nejati *et al.*, 2006] and that learning flat macros e.g. [Mooney, 1990] is more equivalent to the finite state machine approach.

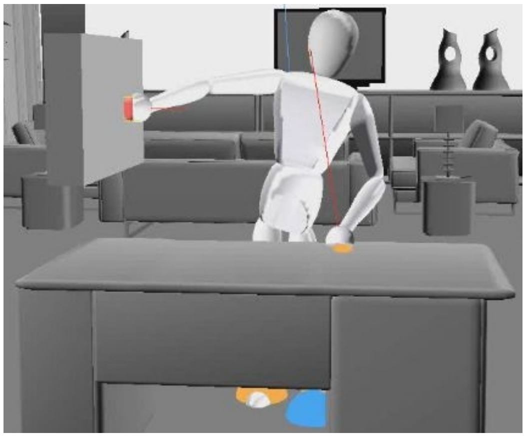

Figure 4: Example WBC behavior for cleaning a vertical surface while maintaining several operational constraints

3 Prioritized Multi-Objective Control: Executing Complex Behaviors

Choosing a symbolic action does not make a robot move its motors. The discrete symbolic structure has to be translated into continuous sensori-motor feedback, and for this we rely on model-based control applicable to mobile manipulators and humanoid robots. We consider the problem of coordinating the physical behavior of the robot operating in the complex environment. The robot is required to accomplish arbitrarily complex tasks, which involve manipulation and locomotion behaviors as well as the handling of environmental constraints.

In particular, for this work we exploit our work on interactive control of a humanoid robot [Sentis and Khatib, 2005]. This framework leverages potential field control techniques to address the simultaneous optimization of multiple low-level criteria characterizing the skills of the robot. It combines the potential fields from all desired criteria using a prioritized control hierarchy, producing motion behaviors such that all criteria can be optimized while satisfying the assigned priorities. Figure 4 illustrates an interactive behavior with multiple potential fields to clean windows. The criteria designed to execute this behavior are shown in table 2.

The difficulty of operating in every-day human surroundings arises from their inherent complexity and variability. Using whole-body skills as described above, we address all aspects of the motion including both goal-based tasks and constrained behaviors. To deal efficiently with constraints we build models that address the complex contact and topological interactions with the environment.

For example, we have recently developed a model called the virtual linkage model to characterize the contact state of the robot. Using optimization techniques it enables the design of internal force behavior and locomotion policies that comply with frictional and rotational contact constraints.

To implement potential field control strategies, we create a generalized dynamic model of the robot that relates actuator and body accelerations to generalized control torques as well as to contact forces with the environment. This model

Table 2: Decomposition for the task shown in figure 4.

Task Primitive	*Coordinates*	*Control Policy*
Contact support	internal forces	optimal contact
Joint Limits	joint positions	locking attractor
Self Collisions	distances	repulsion field
Balance	CoM(x, y)	position
Right hand	Cartesian	force and position
Gaze	head orientation	position
Upright posture	marker coordinates	captured sequences

provides an effective interface to project artificial potential fields into actuator space. Working at the torque level and aided by the dynamic and contact models mentioned earlier, we create force compliant behaviors that are capable of dealing with unplanned contact events and contact variability in the environment.

Another important characteristic of the execution layer is its hierarchical architecture, which is designed to analyze and handle action conflicts by imposing priorities between the control objectives. Priorities are used as a mechanism to temporarily override certain non-critical criteria in order to fulfill critical constraints. To reinforce the planning process, the execution framework estimates at runtime the feasibility of the commanded actions and returns detailed information on the causes of the conflicts. Aided by Icarus, feasibility information is aimed at triggering the replanning process of the robot's behavior, which will result in finding alternative paths that optimize the desired chores.

4 Mobile Manipulation Database

As depicted in figure 1, the role of the mobile manipulation database (MMDB) is to collect information relevant for the interaction between the various components of the system, and to help mediating the data flowing between them. It allows components to retrieve collections of entities matching some search criteria. For example a path planner could request all location nodes and locomotion links along with their associated path costs. However, the MMDB goes beyond "simple" database operations by using a graph-structure that naturally encodes multiple semantic aspects of the world, and providing an event infrastructure that allows components to be notified when certain types of entities are added, removed, or changed.

Related work that influenced the formulation of the MMDB comes from three main sources. The Semantic Spatial Hierarchy of [Kuipers, 2000] is one of the fundamental contributions allowing to ground an abstract topological representation of space in the noisy and uncertain sensori-motor system of autonomous robots. Later work of the same lab (e.g. [Beeson, 2008]) pursues this line of research, with aspects of sensori-motor learning increasing in importance. [Vasudevan, 2008] solves representation of space using a hierarchy of objects and their relationships, in order to support spatial cognition, and gives a good overview of further related work. Concerning planning and execution, the works of [Haigh and Veloso, 1998] and [Veloso *et al.*, 1995] provide system architectures and planner systems that integrate learning through knowledge structures that are interpretable across

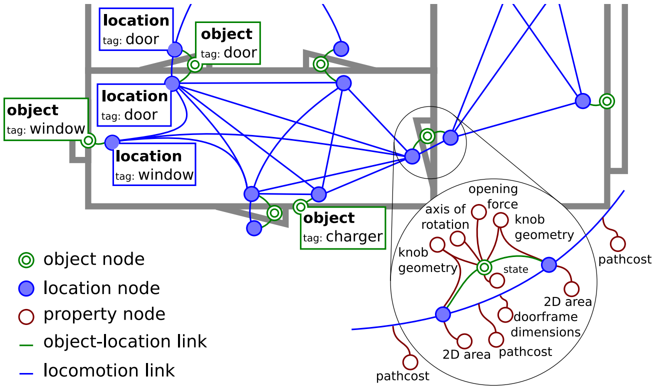

Figure 5: Example of a room with doors, windows, and a charging station, illustrating the graph underlying the MMDB.

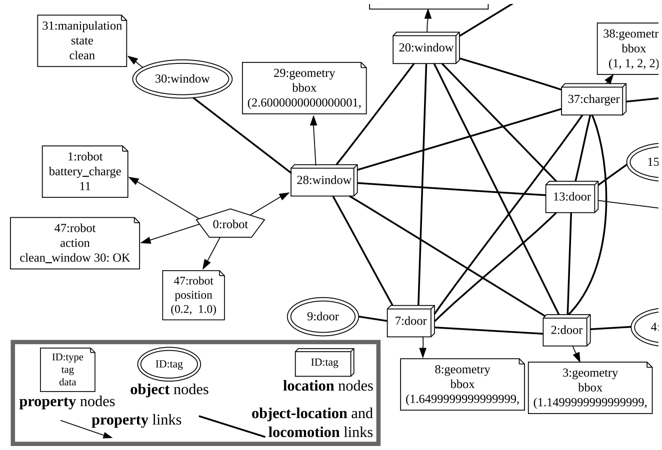

Figure 6: Snapshot of the MMDB (partial view) after the robot has cleaned the window represented by node 30. At this stage, the battery has been depleted to 11% which causes Icarus to interrupt the window-cleaning task in order to go and recharge the robot.

several components. Yet all these contributions are related to mobile robot navigation or HRI and do not directly allow us to integrate manipulation using the whole-body control approach [Sentis and Khatib, 2005].

Intuitively, the MMDB has to *(i)* provide spatial and functional information about the world, *(ii)* represent object identities and connections between regions and objects, and *(iii)* allow components to easily store and retrieve information relevant for their functioning (e.g. "retrieve all locations with links to an object labelled as a cup"). Figure 5 shows an illustrative example.

The MMDB contains *entities* that are either *nodes* or *links*. Each entity has a unique *ID*, an associated *type* and *tag*, and optional mutable *data* which are initialized when the entity is created. **Nodes** represent pieces of information that are relevant for the robot, e.g. an object node representing a cup that it is supposed to wash. **Links** can be directed or bidirectional and represent relationships between nodes, e.g. a collection of SIFT features that the robot can use to detect and localize a cup prior to grasp planning. The **ID** is essential for symbol grounding, it ensures that the various components "talk" about the same "thing", e.g. when Icarus requests locating object 17, the vision system ends up retrieving the correct set of SIFT features in order to find that specific cup. The **type** and **tag** are essential for searching the database and filtering events, they encode an ontology for an application domain. **Data** is usually set only for property nodes, it encodes an actual piece of information, such as a point cloud coming from stereo vision.

We are aware that there is a large body of work on knowledge bases and ontologies that can be exploited for reasoning in the domain of autonomous mobile manipulation. At this early stage of our research, in order to demonstrate that the overall approach is feasible, we chose to use a simple ad-hoc ontology, outlined below. However, given that the *type* of entities is stored as a string, the MMDB remains ontology-agnostic, at least for the time being.

The currently implemented node and link types are: location, object, object-location, locomotion, agent, geometry, and manipulation. **Location** (e.g. door, window, room) permits reasoning about space. **Object** (e.g. door, window, charger) is fundamental for modeling, recognition, and manipulation of objects. **Object-location** links a region of space with a physical entity. **Locomotion** links two regions of space that are known to be reachable via locomotion from each other. **Agent** nodes collect a robot's internal state information as far as it pertains to more than one component. **Geometry** properties (e.g. bounding box, length, point-cloud, mesh) collect metric information about entities, e.g. for grasp planning or fusion over longer exploration periods. **Manipulation** properties (e.g. state, rotation axis, maximum forces) store data necessary for planning or controlling manipulation tasks.

5 Evaluation in an Example Domain

As a prototypical evaluation scenario, we have chosen to implement a world with rooms, doors, windows, and a charger. The task of the robot is to clean all the windows, exploring all rooms and recharging itself as required (all operations use up some amount of energy). Some of the doors are locked, in order to verify that failure detection at the WBC level can be propagated to Icarus. However, integration with WBC is not yet done, so at this stage we verify the symbol grounding of geometrical entities by assigning bounding boxes to objects and simulating controller failures and successes depending on the location of the robot with respect to these bounds (e.g. it needs to be close enough to a window to clean it). When a door to a previously unknown room is opened, the windows and doors contained in the new room are injected into the MMDB. Thus, the controller is not aware of any symbols, Icarus is not aware of any bounding boxes, and the MMDB provides the translation between the two. The result of cleaning a window gets reflected in the *manipulation state* property of window objects, which starts out as *dirty* and transitions to *clean* only if the controller succeeded. Similarly, doors can be in an *open*, *closed*, *locked*, or *unknown* state.

Icarus is a Lisp program and the MMDB has been prototyped in Python. The two are connected using XMLRPC, and at each step the entire graph is logged, yielding output similar to figure 6 (after some manual layout adjustments). This example demonstrates how MMDB translates between symbols and geometric entities, how Icarus makes skill selection flexible and scalable (adding door opening skills and exploration goals are simple additive adjustments that do not require rewiring any program) and how the window-cleaning task can be interrupted by battery-charging and then resumed.

6 Conclusions & Outlook

In this position paper we have presented early stages of our work on a mobile manipulation database and have motivated this research in the context of autonomously performing tasks that can be useful in every-day human environments. Symbolic representations and reasoning methods on the one hand allow to make globally informed decisions and ensure that execution is goal-oriented, by "abstracting away" much of the low-level details in order to make the problem tractable. On the other hand, continuously controlling the motion of an autonomous robot such that it safely and effectively operates in physical contact with objects and even humans focusses on more local details such as real-time control and rigid body dynamics.

The evaluation we presented demonstrates that two of the three objectives mentioned in the introduction are presently fulfilled by the MMDB: *(i)* ground symbols in the robot's low-level sensing and action capabilities, *(ii)* maintain and share information relevant to more than one component. Concerning our objective to *(iii)* support life-long adaptability of the robot in changing and uncertain environments, we give a motivating example as outlook: Suppose that the perceptual apparatus can detect doors but does not provide information needed for operating it, such as whether it is a sliding or rotating door, or how exactly to grip the handle. WBC provides sensori-motor exploration to figure out the missing details by trying out various alternatives. Once the robot has discovered how a particular door can be opened, the relevant data is stored in the MMDB, attaching this WBC-specific information to an entity that is shared between all components. Later, the robot perceives another door of similar appearance. We can then use the parameter set of the first door as an initial guess at how this new door can be opened.

The MMDB provides representational support for system integration, aimed at fulfilling a specific set of requirements for a more general problem, namely to expose those parts of a component's internal data structures that are needed for useful interaction between approaches coming from various sub-fields of AI and robotics. In this position paper, we only present a prototypical implementation of the MMDB, which lacks the event mechanisms that shifts the burden of detecting environment dynamics and the action of other components into a representational system that has all the required information at its disposal. For the proof-of concept, the lack of events is not a limiting factor: the number of nodes and links is very small, and events are emulated by iterating over all entities at each step. However, it is already apparent that these capabilities will promote loose but effective coupling between components. In particular, we are interested in adding learning at the HTN level, exploration and SLAM at the geometric and spatial-topological levels, and providing a rich set of sensory inputs.

One promising direction of future research concerns feeding high-level contextual information from Icarus into the MMDB. We expect this to help with disambiguating object perception, and to provide more fine-grained specification of WBC behaviors. For example, it would be possible to inject or relax manipulation constraints such as maintaining an upright orientation of a container depending on whether it contains liquid or is empty. Exploration of unknown environments is another application that we feel will benefit from the MMDB.

References

[Beeson, 2008] Patrick Beeson. *Creating and Utilizing Symbolic Representations of Spatial Knowledge using Mobile Robots*. PhD thesis, The University of Texas at Austin, August 2008.

[Choi and Langley, 2005] Dongkyu Choi and Pat Langley. Learning teleoreactive logic programs from problem solving. In *Proceedings of the Fifteenth International Conference on Inductive Logic Programming*, pages 51–68, 2005.

[Haigh and Veloso, 1998] Karen Zita Haigh and Manuela M. Veloso. Interleaving planning and robot execution for asynchronous user requests. *Autonomous Robots*, 5(1):79–95, 1998.

[Kuipers, 2000] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[Langley and Choi, 2006] Pat Langley and Dongkyu Choi. A unifed cognitive architecture for physical agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 51–68, 2006.

[Mooney, 1990] R. J. Mooney. *A general explanation-based learning mechanism and its application to narrative understanding*. Morgan Kaufmann, San Mateo, CA, 1990.

[Nau *et al.*, 2003] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, W. Murdock, D. Wu, and F. Yaman. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

[Nejati *et al.*, 2006] Negin Nejati, Pat Langley, and T. Konik. Learning hierarchical task networks by observation. pages 665–672, 2006.

[Nilsson, 1994] Nils Nilsson. Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research*, 1:139–158, 1994.

[Sentis and Khatib, 2005] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518, 2005.

[Vasudevan, 2008] Shrihari Vasudevan. *Spatial Cognition for Mobile Robots: A Hierarchical Probabilistic Concept-Oriented Representation of Space*. PhD thesis, ETH Zürich, Switzerland, 2008. Diss. ETH no. 17612.

[Veloso *et al.*, 1995] Manuela M. Veloso, Jaime Carbonell, Alicia Pérez, Daniel Borrajo, Eugene Fink, and Jim Blythe. Integrating planning and learning: the prodigy architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):81–120, 1995.