FP6-004381-MACS

MACS

Multi-sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances

Instrument: Specifically Targeted Research Project (STReP)

Thematic Priority: 2.3.2.4 Cognitive Systems

D4.4.4 Submission of a conference or journal article describing the results of D4.3.4 and D4.4.3

Due date of deliverable: November 30, 2007
Actual submission date: November 27 / December 3, 2007

Start date of project: September 1, 2004 Duration: 39 months

University of Osnabrück (UOS)

Revision: Final

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

EU Project MACS

Deliverable D4.4.4

Submission of a conference or journal article describing the results of D4.3.4 and D4.4.3

Christopher Lörken, Joachim Hertzberg

Number: **MACS/4/4.4**
WP: *2*
Status: *final*
Created at: *December 3, 2007*
Revised at: *v1 – December 3, 2007*
Internal rev: *v1 – December 3, 2007*

FhG/AIS	Fraunhofer Institut für Intelligente Analyse- und Informationssysteme, Sankt Augustin, D
JR_DIB	Joanneum Research, Graz, A
LiU-IDA	Linköpings Universitet, Linköping, S
METU-KOVAN	Middle East Technical University, Ankara, T
OFAI	Österreichische Studiengesellschaft für Kybernetik, Vienna, A
UOS	Universität Osnabrück, Osnabrück, D

This research was partly funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.

Corresponding author's address:

Christopher Lörken
Universität Osnabrück
Institut für Informatik
Albrechtstr. 28
D-49076 Osnabrück, Germany

IAIS Fraunhofer Institut Intelligente Analyse- und Informationssysteme	Fraunhofer Institut für Intelligente Analyse- und Informationssysteme Schloss Birlinghoven D-53754 Sankt Augustin Germany	Tel.: +49 (0) 2241 14-2683 (Co-ordinator) **Contact:** Dr.-Ing. Erich Rome
JOANNEUM RESEARCH	Joanneum Research Institute of Digital Image Processing Computational Perception (CAPE) Wastiangasse 6 A-8010 Graz Austria	Tel.: +43 (0) 316 876-1769 **Contact:** Dr. Lucas Paletta
LINKÖPINGS UNIVERSITET	Linköpings Universitet Dept. of Computer and Info. Science Linköping 581 83 Sweden	Tel.: +46 13 24 26 28 **Contact:** Prof. Dr. Patrick Doherty
Middle East Technical University	Middle East Technical University Dept. of Computer Engineering Inonu Bulvari TR-06531 Ankara Turkey	Tel.: +90 312 210 5539 **Contact:** Asst. Prof. Dr. Erol Şahin
Ö F A I	Österreichische Studiengesellschaft für Kybernetik (ÖSGK) Freyung 6 A-1010 Vienna Austria	Tel.: +43 1 5336112 0 **Contact:** Prof. Dr. Georg Dorffner
UNIVERSITÄT OSNABRÜCK	Universität Osnabrück Institut für Informatik Albrechtstr. 28 D-49076 Osnabrück Germany	Tel.: +49 541 969 2622 **Contact:** Prof. Dr. Joachim Hertzberg

1 Document contents

The attached article titled “Grounding planning operators by affordances” has been submitted on November 27, 2007, to the CogSys 2008 conference, http://www.cogsys2008.org/index.php. The submission has been confirmed by the conference organizers:

Message from The PaperPlaza Conference Management System

Dear Mr. Christopher Lörken,

This email is to acknowledge receipt of your submission to the 2008 International Conference on Cognitive Systems (CogSys 2008) to be held during April 2-4, 2008 in University of Karlsruhe, Karlsruhe, Germany.

Attached below are pertinent data on your submission for you to verify. If there is any error or necessary change to your contact data, please update the data at the following site:

http://ras.papercept.net

If you need to modify the title or author list, you will have an opportunity to do so when submitting the final version for publication in the proceedings.

For any correspondence regarding this paper, please be sure to refer to your paper ID.

Tamim Asfour
Program Chair, CogSys 2008

Grounding planning operators by affordances

Christopher Lörken and Joachim Hertzberg
University of Osnabrück, Institut für Informatik
D-49069 Osnabrück, Germany
{cloerken | hertzberg}@informatik.uni-osnabrueck.de
http://www.inf.uos.de/kbs

Abstract— **This paper introduces a view on planning that is based on interpreting objects not by their identity but by their functionality. Ecological psychologist J.J. Gibson's concept of *affordances* provides the underlying theory for developing a system that does not distinguish between objects of functional equivalence. It will be argued that the plans resulting from this approach are both less complex and still more flexible and robust in their application while proposing solutions to symbol grounding of objects in the environment and planning operators in execution behaviors. The system will be exemplarily presented.**

Keywords: planning, affordance, functional equivalence, symbol grounding, autonomous robot, real-world application

I. MOTIVATION

Controlling an autonomous robot has been among the chief target applications for the field of action planning in AI, starting with STRIPS/SHAKEY [19]. Planning has made considerable progress in the recent past [8], and so has robotics. Then why aren't smart plan-guided robots all over the place?

A fundamental problem in plan-based robot control, if "plan" is understood to be a state-of-the-art AI plan, is that it would require to solve a symbol grounding [10] problem in two directions. First, the objects and relations that occur in the domain need to be identified from the robot's sensor data; e.g., in order to move box_A from room X to Y, it needs to be identified and distinguished from, say, box_B and box_C. While this may work well for certain objects, such as for a neighboring room that gets identified by way of robot self-localization in a prior map, it may be very expensive to do for others. Second, the abstract operators contained in a plan need to be grounded in actual control rules or behaviors of the robot. While this has been tackled in several versions in hybrid robot control architectures [2], it is still far from being well-understood.

This paper explores an approach to conceptualizing differently the entities that a plan-based robot perceives in its environment and, hence, the way in which operators are grounded in physical robot behavior. It is based on the notion of *affordances*, as developed by J.J. Gibson. His idea was that biological species would perceive important action potentials of their direct environment on an elementary level: e.g., the affordance of "something" in the environment to be lifted by some agent can be perceived directly without prior object segmentation and classification; it may be triggered by a rock and a glass and a cell phone all alike for, say, a human-sized agent. We will go into some detail in Sec. II. Note that we do not postulate that agent and robot perception has to be based on affordances alone: individual objects, e.g., may play a role in addition, and anchoring [4] may be a method of grounding them in sensor data. We focus on affordances in this paper to make clear the potential that they offer for plan-based robot control.

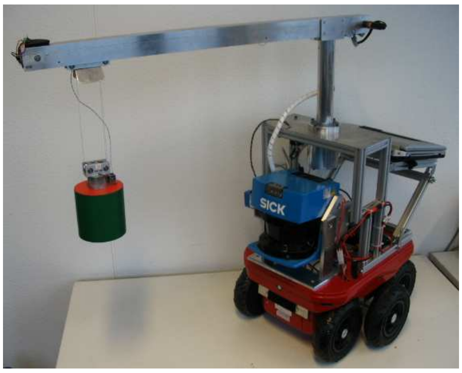


Fig. 1. The Kurt3D robot with magnetic crane manipulator. Picture courtesy of Andreas Nüchter and Frank Meyer.

And here is this potential in a nutshell: Assume a robot is able to perceive some affordance "directly", i.e., by very cheap algorithms on its sensor data, which do not require object segmentation, classification and reasoning. Assume further that the physical behavior, like lifting, linked to the perceivable affordance is modeled abstractly as an operator in the planning domain description. Then grounding this abstract operator in physical robot behavior would mean to guide the robot into a situation where the sensor data known to trigger the affordance is known or expected to be present, and then rely on the affordance-behavior coupling to guide the physical behavior. So if the robot perception process does include perception of affordances (their number may be large or small), then tailoring the plan operators "in the right way" according to these affordances will solve the grounding problem in both directions. This will be detailed

This research is funded by the European Commission's 6th Framework Programme IST Project MACS [16] under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.

and exemplified in the body of the paper.

Let us emphasize that the current paper's contribution lies in the way in which domains with affordances present are modeled for planning and in grounding operators. The planning *process* and *algorithms* do not change: In fact, we will show in the body of the paper that a standard planning system (FF in this case) based on a standard planning domain description in terms of PDDL can still be used, making it possible to profit from latest planning technology. A blend of planning and situatedness has been an issue for long, Agre and Chapman's work on PENGI [1] being a prominent exemplar. Yet our approach differs from that line of work in that we keep the classical planning representation and algorithms, tailoring them such that, by aid of affordances, particular actions in the classical plan may physically run as situated behaviors; the earlier work, on the other hand, has put forward that situated plans be entirely different in nature from classical ones.

The paper is structured as follows: Section II will describe and interpret the affordance term and start to address its meaning and usefulness for robotics. Section III will describe the functionality and underlying thoughts of the planning system and of the closely related execution module and shed light on how operators can actually be grounded by affordances. Sections IV and V will describe the developed system including the necessary representations and will present exemplary results before the whole paper will be wrapped up and discussed in section VI.

II. AFFORDANCES

The term affordance has been coined by ecological psychologist J.J. Gibson almost 30 years ago. He states:

> “The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill.“ [9, p. 127]

To skip some decades of vivid discussion and different views of Gibson's concept we will follow the interpretation proposed by Anthony Chemero [3].[1] Chemero describes an affordance as a perceivable relation between an environmental object and an agent that describes some potential for action. In other words, an affordance describes what the agent can do with the object. A stone might, e.g., afford to be thrown or to be used as a paperweight. The affordance itself is hereby subjective to the agent, i.e. the same stone might afford to be liftable for a strong agent, but only afford sitting on it for a smaller, weaker agent.

We thus interpret affordances as directly perceivable mappings of environmental features or characteristics to the abilities for interaction of the particular agent, or in our case of a robot. We therefore see affordances like *liftable*, *pushable*, or *passable* that a robot can perceive in its environment. A robot as the one shown in Figure 1 that has a magnetic crane as its primary manipulator will, e.g., perceive magnetic objects below a certain size and with a fairly flat top as affording the lift action.

Gibson's notion of affordances can thus be seen as a function-centered view on perception. When speaking of affordances one assumes a system that interprets its surroundings by what it can do with it, how it can interact with it, instead of what things are and if they can be recognized. In other words, we do not need to detect and categorize objects in order to know what to do with them; thus solving the bottom-up problem of symbol grounding. We can furthermore formulate affordance-related operators like, e.g., $lift(liftable)$ that can, in a top-down manner, be grounded on a physical behavior that is being guided by the perceived affordance to actually lift something that affords to be lifted.

In the following section, we will now describe how standard planning techniques can be enriched by incorporating the affordance concept and how such a planning system and a robot's execution control complement each other and benefit from this extension.

III. PLANNING WITH AFFORDANCES

We are aiming at a system that embeds the expressive power of the affordance concept in standard planning techniques to show that we provide an additional way of representing planable knowledge that does not conflict with existing standards. The approach that we are going to introduce will thus, in the end, use the well-known and established planning tools of the Planning Domain Definition Language (PDDL, [17]) and the Fast-Forward (FF) planner by Jörg Hoffmann [13].

The overall system operates on a knowledge representation or world model based on a coarse symbolic map defining some regions in space (see Figure 2(b)). In contrast to other current systems we are not inserting objects with associated feature lists in that map but instead *affordance labels*. That means that we assume our robot to explore the environment and perceive, e.g. the affordance of a liftable item.[2] Then, the map-region in which it has perceived that affordance will be tagged liftable.

If we now recall that the intrinsic meaning of the affordance concept allows us to actually specify an operator that says $lift(liftable)$ with $liftable$ coding for any suitable object that the robot actually can lift the simplification of this approach in comparison to other planning techniques becomes evident. Take the simple example that all blue objects are liftable for the robot and that all red objects with a flat surface are liftable as well. This leaves us with two actual instantiations of the affordance liftable. If the planner now does not care about these instantiations but instead gives the operator $lift(liftable)$ to the execution control of the robot, the robot can look in its memory what particular instantiations of the affordance liftable it knows (namely exactly those two) and configure its perception to

---

[1] For a wrap-up of the discussion and an elaboration on the use of affordance theory for robotics see, e.g., [14].

[2] How affordances are being perceived and represented will be topic in the next section.

look for the cues that are connected with those instantiations. In this case, the robot would look either for blue color blobs or for red and flat structures and can then ground the $lift$-operator on whatever cues it perceives first that meet these requirements. It is neither necessary to recognize objects nor to specify how the operator is being grounded since it is only at execution time when the robot has to decide with which behavior and environmental object or percept the operator will be implemented.

This has the following large benefits:

1) The search space for the planner to explore is significantly reduced. Instead of dealing with a potentially large number of operator instances built from all liftable objects (the way in propositional planning), or of representing all these objects by a variable and employing a more complex least-commitment planner (the way in classical partial order planning), we are using just one affordance at planning time.
2) The system will be more flexible in execution as the first appropriate object will be selected in the current environment; be it static or dynamic.
3) The actual plan operators do not have to be changed if the robot might get another class of objects that are liftable, e.g. cylinders, since the notion of categorized objects is avoided in the planner. It will only be needed to add another affordance instantiation telling the system that cylindric objects are liftable as well. Then they will be selected automatically together with the other instantiations for grounding the $lift$-operator during execution.

It is to note here that we do not aim at a system that completely avoids the notion of objects. In fact, quite the opposite is the case: objects are very valuable and needed for high-level planning and we think that everything that is worth to be anchored as an object should in fact be anchored as an object. But the concept of affordances allows us to neglect the actual object term and the connected workload in generating and maintaining their representation in the world model as well as their additional load on planning for exactly those items where we do not need the actual identity of that very item. I.e. when we do not care if we lift box_A or box_B or even $cylinder_{42}$.

In the following document, we will, nevertheless, focus on a scenario where we do not need the representation of objects at all in order to demonstrate the power and usefulness of the affordance concept. An overall system should of course aim at combining the following affordance extension with standard planning approaches.

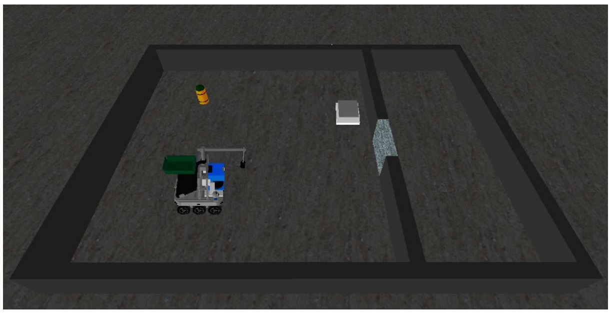
Section IV will now describe in a bit more detail the underlying representation of both affordances and the general working domain before section V will present the plans that result from this representation.

IV. MODELING AFFORDANCES FOR PLANNING

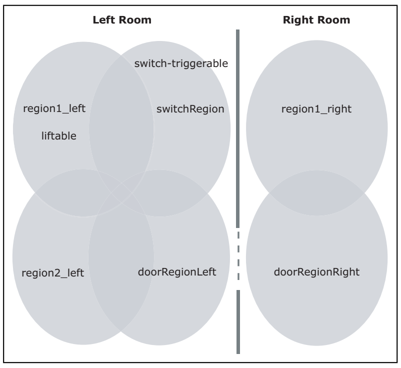
A. Affordance Representation

Being a relational property of both the environment and the robot, affordances are represented here in a triplet form. The so-called affordance representation triple (ART) has the structure:

$$\left\langle \begin{array}{c} cue\ descriptor, \\ behavior\ descriptor, \\ outcome\ descriptor \end{array} \right\rangle \tag{1}$$



(a) Demonstrator scenario.



(b) Symbolic map.

Fig. 2. Scenario and map. Coarse localization in soft regions is sufficient for affordance-based planning. The scenario picture is taken from the MACSim simulator by METU-KOVAN [5].

The *cue descriptor* part contains information like, e.g., color or shape while the *behavior descriptor* describes the action the robot actually performed and the *outcome descriptor* describes the effect of that action. Staying with the example of the affordance liftability, a *liftable* ART may thus easily represent that all blue objects are liftable, e.g. with a crane or an arm. The actual ARTs may either be handcoded or acquired during an exploration phase of the system in which the robot tries out its different actions on various environmental objects (cf. e.g. [7], [21]).

Let us assume here that the robot is able to detect the cues described in the different cue descriptor parts of its learned ARTs within its environment since the details regarding these procedures are out of scope for this paper (cf. e.g. [20]).

Note, however, that we distinguish the *abstract affordance type* $liftable$ from its different instantiations, the ARTs, that belong to that type. So the two different instantiations mentioned in the previous section of blue or red and flat liftable objects correspond to two distinct affordance representation triples of the same abstract affordance type liftable.

B. Experiment Setup and World Model

We chose a simple demonstration scenario consisting of two rooms, a separating wall with a door, and a switch that has to be weighted for the door to open (see Figure 2(a)). To represent this setup and the current state of the world in a form understandable by the planner, we introduce the map depicted in Figure 2(b). This symbolic map divides the room structure into 6 labelled regions whereby the regions are not meant to be separated sharply but may blur into each other.

The content of this map that has to be maintained by the system is whether an affordance has been perceived in one of the regions. Most importantly, it only tags the map region with a label stating that it has perceived that abstract affordance type, e.g. $liftable$ in Figure 2(b), within that region; not with the actual ART whose cues it has perceived and neither with the presence of some cues in that area. The map is concurrently being updated while the robot is moving through its environment.

C. Domain Description

Having introduced the representation of affordances and the underlying symbolic world model allows to move on to the actual planning side of the system. As aforementioned, the planning system itself is completely modeled in PDDL using the standard FF-planner for planning.

The domain description, that will for reasons of space be only provided partially here, does not differ much from a common PDDL description. However, we have added affordance-predicates that reflect the truth value of whether an abstract affordance type has been perceived within a particular region of the map.

Listing 1 Predicates

```
(:types region room)
(:predicates
    (robotAt ?region - region)
    (inRoom ?region - region ?room - room))
    (hasLiftedSomething)
    (liftable ?region - region )
    (switch-triggerable ?region - switchRegion)
    (passable ?startRegion ?targetRegion - region))
```

Listing 1 defines next to some simple world fact predicates the atomic formulas $liftable$, *switch-triggerable*, and *passable*. Each of these encodes the robot's percept of an according affordance type within a particular map-region. So, e.g., $liftable(region2\_left)$ would be contained in the situation description (see listing 4), if the robot has previously perceived the cues of one or more affordance representation triples belonging to the abstract affordance type liftable. *switch-triggerable* relates to the robot's impression that it can put something on the switch while *passable* describes the affordance of something like a gap in a wall that can be passed through. These predicates thus code for the general impression of the robot to have perceived some affordance within a region.

Due to the nature of the affordance concept one can use this symbolically encoded knowledge about the possibility for action a certain area provides to model operators that make use of this knowledge by using it as a precondition.

Listing 2 Lift - Action

```
(:action lift
  :parameters (?region - region)
  :precondition
    (and
      (robotAt ?region)
      (liftable  ?region)
      (not (hasLiftedSomething)))
  :effect
    (and
      (hasLiftedSomething)
      (not (liftable ?region))))
```

Listing 2 thus shows a lift operator that has in its precondition to be in a region where the liftability affordance has been perceived.

If that operator is being reached during execution, the robot will retrieve all liftability ARTs from its memory, configure its perception to look out for the cues represented in the cue-descriptor of the ARTs and eventually select a suitable liftable object within that region to lift it. It will ground the $lift$-operator elegantly on the next best object that actually affords lifting to the robot and on the appropriate behavior to lift it.

Listing 3 shows another example for an affordance-based operator, namely $trigger\text{-}switch$, that is insofar different from the last example as it contains the assumption of a perceivable affordance on the effect or outcome side of the operator.

Listing 3 Trigger-Switch - Action

```
(:action trigger-switch
  :parameters (?doorRegion          - doorRegion
               ?otherDoorRegion     - doorRegion
               ?switchRegion        - switchRegion)
  :precondition
    (and
      (robotAt ?switchRegion)
      (hasLiftedSomething)
      (switch-triggerable ?switchRegion)
      (not (= ?doorRegion ?otherDoorRegion)))
  :effect
    (and
      (passable ?doorRegion ?otherDoorRegion)
      (passable ?otherDoorRegion ?doorRegion)
      (not (switch-triggerable ?switchRegion))
      (not (hasLiftedSomething))))
```

This is of course needed for propper planning precisely because a closed door does *not* afford to be passable. But as that means that passable would not be contained in the map since it cannot be perceived by the robot the system needs the knowledge of how to derive that fact or generally speaking of what affordance might be perceivable as the result of an action.

This is in some way contradictory to the original affordance idea of Gibson, who claimed that affordances are immediately there to be perceived and that one does not reason about them. While this view, due to the computational complexity of perception, has already been questioned in robotic literature (cf. e.g. [15], [14]) it does not even demand more expert knowledge than needed in standard object-feature based planning systems. Such a system would simply model that the door-object has, as a result of the trigger-switch operator, the feature to be open and that feature would in turn be a precondition for the operator drive-through.

D. *Different Types of Operators*

As we have already stated above, the affordance concept offers a valuable way to extend existing planning approaches with operators that make use of the perceivable functional equivalence of objects. The introduced system is, however, really an extension, not a substitution. Even in this simple example scenario there are operators where it does not actually make sense to formalize them as depending on affordances. In this case, these operators are connected not to objects, but to the basic navigational skills of the robot. Above it has always been assumed that the robot already was in a particular region in order to demonstrate the affordance-based operator grounding. Of course one could argue that flat floor affords navigability for the robot; and in fact there are some approaches focussing on affordance-use for navigation (e.g. [6], [18], [21]). In our case, where each region in the same room is approachable from each other region, an *approach*-operator does, however, neither need to be affordance-based nor would it be beneficial since adding affordance percepts for the navigability to each region in the map does not contain any valuable information whatsoever (cf. the more general discussion on this matter in [12]).

V. RESULTS

A. *Planning*

During the last section, excerpts of the developed affordance-based domain description have been provided that can be used as the input to an off-the-shelve planner.

Listing 4 shows an exemplary problem definition for this domain that specifies the goal for the robot to get into the right room. In order to make things interesting, the door between the two rooms is closed but the robot has already perceived the liftable affordance in region2_left and the affordance of a triggerable switch in the switch_region.

Listing 5 shows the resulting plan as it has been generated by the FF-planner omitting some implementation specific parameters. As one can see, the plan tells the robot to move to the region where it knows that something liftable is present. Then, in that region, the plan tells to just lift up *something liftable*, to carry it to the switch, open the door, and eventually act upon the passable affordance to drive to the other room.

Listing 4 Problem Definition

```
(define (problem macs-prob)
   (:domain macs-example)
   (:objects
      region1_left region2_left region1_right - region
      switchRegion - switchRegion
      doorRegionLeft doorRegionRight - doorRegion
      rightRoom - room
      leftRoom   - switchRoom )
   (:init
      (robotAt  region1_left)
      (inRoom  region1_left         leftRoom)
      (inRoom  region2_left         leftRoom)
      (inRoom  switchRegion         leftRoom)
      (inRoom  doorRegionLeft       leftRoom)
      (inRoom  region1_right        rightRoom)
      (inRoom  doorRegionRight      rightRoom)
      (liftable  region2_left)
      (switch-triggerable  switchRegion))
   (:goal
      (robotAt  region1_right)))
```

Listing 5 FF Generated Plan

```
0: APPROACH-REGION    region2_left
1: LIFT               region2_left
2: CARRY              switchregion
3: TRIGGER-SWITCH     switchregion
4: APPROACH-REGION    doorregionleft
5: CHANGE-ROOM        doorregionleft   doorregionright
6: APPROACH-REGION    region1_right
```

B. *Execution*

Given a plan as in Listing 5, the execution module only has to sequentially select the operators and to execute them step by step. In case of the type of non affordance-related operators, as e.g. *approach*, the system's execution module can simply go on by triggering whatever control routines are implemented to carry out the desired action. If the operator is affordance-related, i.e. if it has the percept of an affordance as precondition, the execution module has to retrieve all affordance representation triples that belong to the affordance type specified in the operator's precondition and to configure the robot's perception to look out for their cues. It so selects the next best object that fulfills the required cues and grounds that operator on that object by using it for interaction as specified in the behavior descriptor. Hereby, it does neither matter which particular object it is, be it box_A or box_B nor which particular ART has been detected, be it the one for liftable boxes or the one for liftable cylinders.

The resulting plan execution is thus more flexible than the one based on object-centered domain representations. The reason is that the same operator may get grounded in multiple physical actions in the environment, depending on which box gets chosen at execution time. The choice emerges from the actual interplay of perception and behavior during execution; given that there is no need to distinguish between different objects offering the affordance in question, this non-commitment is perfectly acceptable. The execution has a strong flavor of opportunism [11] in the sense that it will jump to opportunities that may not even have been present at planning time. The region model proposed for the knowledge representation does furthermore add robustness to the system since the robot just moves to a coarse region where it had the impression for the desired action to be afforded. If the objects that resulted in the percept of the affordance have moved within that region, because of some environment dynamics, the execution will not even take notice as the robot will simply interact with the next available and suitable object.

VI. Summary and Discussion

The presented system proposes a new view on operator grounding as an extension to standard planning approaches. Objects are represented not by their identity but by the abstract kind of interaction possibility these objects afford to an agent or robot, i.e. by their affordance relation.

A symbolic formalization of this concept has been introduced using only the common tools of the modeling language PDDL and results have been presented with a not customized FF-planner.

Combined with the introduced way of grounding operators during the execution phase the resulting plans constitute a simplification and narrowed search space compared to propositional planning or classical partial order planning approaches.

It has furthermore been argued that the resulting system is flexible and robust in terms of plan execution as the interaction objects are not predefined but instead selected and grounded in a behavior during the execution phase.

The system scales easily with newly discovered objects since they only necessitate new affordance representation triples to be learned but no change of the planner at all. Facing a physically different robot, the planning system has only to be adopted if that robot can achieve qualitatively different actions. That means that there is no need for alterations if the robot, e.g., simply uses an arm to lift things instead of a crane. What changes is only the interpretation of the affordance of liftability, i.e. the actually ART that would contain different cues for liftable objects for a robot with a crane or an arm. New operators only have to be designed for outcomes the old robot wasn't able to achieve, e.g. throwing with an arm manipulator.

One of the most interesting next steps is to reintroduce the notion of objects to the affordance-based planning approach. As stated above the presented system is meant to augment the existing approaches as it tries to limit some of the complexity today's planning systems have to cope with. It is, nevertheless, crucial for a valid planner to be able to actually distinguish similar objects since it is in fact sometimes of concern not to use the expensive vase to trigger the switch. The interesting question when combining the systems is thus where and how to draw the line between those objects that need to be anchored and identified and those that can be interpreted via their functional equivalence.

A second interesting question is how to formally distinguish operators that are meaningfully affordance-related from those that are not. We believe that this operator – affordance relation is meaningful exactly in those cases when an operator can be grounded by exploiting the affordance. And we believe that these are always those operators that directly describe an action; like e.g., lifting a box but not attending a meeting or getting coffee. While coffee affords drinking going to the coffee machine in order to get it is just a necessary planning step before the affordance can be exploited. The means for formalizing this interpretation do still have to be evaluated.

References

[1] P. Agre and D. Chapman. Pengi: An implementation of a theory of action. In *Proc. AAAI-87*, pages 268–272. Morgan Kaufmann, 1987.

[2] R. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[3] A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.

[4] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Auton. Syst.*, 43(2-3):85–96, 2003. Special issue on perceptual anchoring. Online at http://www.aass.oru.se/Agora/RAS02/.

[5] E. Şahin. Middle East Technical University, Department of Computer Engineering, 2007. http://www.kovan.ceng.metu.edu.tr.

[6] P. Doherty. Linköpings universitet, Department of Computer and Information Science, 2007. http://www.ida.liu.se.

[7] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner. Learning predictive features in affordance-based robotic systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 3642–3647, Beijing, China, October 9-15 2006.

[8] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[9] J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Hillsdale, 1979.

[10] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[11] B. Hayes-Roth and F. Hayes-Roth. A cognitive model of planning. *J. Cogn. Sci.*, 3:275–310, 1979.

[12] J. Hertzberg, K. Lingemann, C. Lörken, A. Nüchter, and S. Stiene. Does it help a robot navigate to call navigability an affordance? In *Affordance-Based Robot Control*, Springer LNAI, 2007. (accepted).

[13] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

[14] C. Lörken. Introducing affordances into robot task execution. In *Publications of the Institute of Cognitive Science (PICS)*, volume 2-2007. University of Osnabrück, Osnabrück, Germany, May 2007. ISSN 1610-5389.

[15] K. F. MacDorman. Grounding symbols through sensorimotor integration. *Journal of the Robotics Society of Japan*, 17(1):20–24, 1999.

[16] MACS. Webpage, 2004. http://www.macs-eu.org.

[17] D. McDermott. PDDL - The planning domain definition language. Technical report, Yale University, 1998.

[18] R. R. Murphy. Case studies of applying Gibson's ecological approach to mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):105–111, January 1999.

[19] N.J. Nilsson. Shakey the robot. Technical Report TN 323, SRI International, April 1984.

[20] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner. Learning to perceive affordances in a framework of developmental embodied cognition. In *Proc. International Conference on Development and Learning, ICDL 2007*, London, UK, July 11-13 2007.

[21] E. Uğur, M. R. Doğar, M. Çakmak, and E. Şahin. The learning and use of traversability affordance using range images on a mobile robot. In *Proceedings of IEEE Intl. Conf. on Robotics and Automation (ICRA 07)*, pages 1721–1726, Rome, Italy, April 2007.