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

D5.4.5 Outlook towards affordance usage observation and imitation

Due date of deliverable: May 31, 2007
Actual submission date v2: July 16, 2007

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

Österreichische Studiengesellschaft für Kybernetik (OFAI)

Revision: Version 1

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 5.4.5

Outlook towards affordance usage observation and imitation

Georg Dorffner, Jörg Irran, Florian Kintzler

Number:	**MACS/5/4/5**
WP:	5.4
Status:	Version 1.0
Created at:	July, 2007
Revised at:	July 15, 2007
Internal rev:	

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:

Georg Dorffner, Jörg Irran, Florian Kintzler
Österreichische Studiengesellschaft für Kybernetik (ÖSGK)
Freyung 6
A-1010 Vienna, Austria

Fraunhofer IAIS 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

Contents

1 Introduction

Within this document mechanisms are described how the developed architecture for learning affordances by self experience, described in Deliverables D5.3.1 [DIKP05], D5.3.2 [DIK06] and D5.3.3 [DIK07], can be extended to gain the capability to observe affordance usage by other agents and to learn affordances by imitation.

Learning by imitation is a complex problem that is unlikely to be solved by a single algorithm. However the application of several mechanisms, all focusing on imitation and observation and a sound basis of self experienced knowledge about the environment and the consequences of applying its own behaviours to the environment seams to be a promising approach for enabling an agent to extend its knowledge to perceive and to use affordances.

The approaches described in this document presume, that an agent already gained knowledge about its own behavioural possibilities, *outcome events* caused by applying these behaviours to the environment, and knowledge about the *cues events* that indicate to which outcome the application of a behaviour in the current situation would lead (see section 2).

The *outcome events* learned by the agent include the description of what happened in the environment (external sensors) and include a description of how the state of the agent changed during the application of one of the agents behaviours. The knowledge about changes in the environment, is the basis on which the system is expected to be able to detect changes that are caused by other acting agents. Based on these observations, the agent can learn to perceive new cues and outcomes and even learn to perceive intended movements of other agents.

2 Prequel

2.1 Affordance Learning

The reason why some building blocks can be used as a *stacking base* or a *stacking element*, while others can only be used as a *stacking top* is among features like weight especially the structure of the top and bottom surfaces of the blocks. Whether two objects can be stacked or not dependents on the top region surface of the element that should provide the base, and depends on the bottom region surface of the element that should be stacked on this base. The two surfaces must be in a certain relation to each other for a successful stacking trial. In simple cases the necessary complementary shape is given all-over the top and bottom region. More complicated objects may only share some of those complementary regions, but at least enough to keep a stacking element grounded on the base.

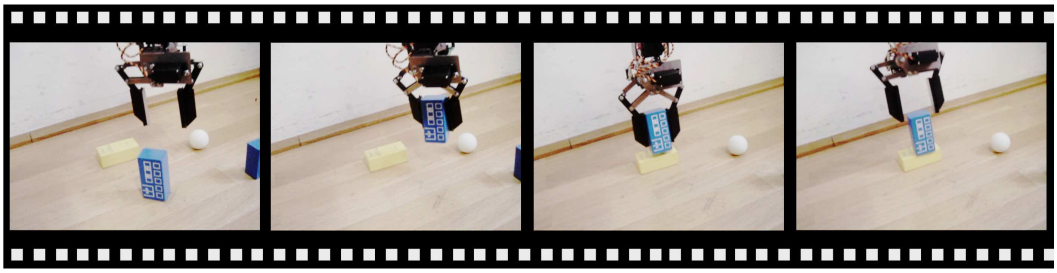

Figure 1: The agent Kurt3D, used in the MACS project, while performing a stacking trial. Thereby the blue and the yellow object can be used as stacking tops and stacking base, while the white ball can not be used as a stacking base. The reasons for this discrimination are the top and bottom surfaces of the objects, which can for the most part not be directly perceived by the agent.

When objects are provided to an agent, the relevant surfaces can't be perceived directly. Nevertheless humans are able to assume whether an object is stackable or not, without seeing this surface. Therefore they use several cues based on their own experience to fulfil this task.

The learning approach presented in Deliverables D5.3.1 and D5.3.2 enables an artificial agent to derive a set of *cue events* as well as knowledge about *outcome events* in relation to its own capabilities, which enables an artificial agent to build hypothesis about the affordances in its environment (in the used example: stackable or not stackable).

In case an agent is supposed to lift objects, the central physical property of the objects, that in one case leads to lift-ability and in the other to non-lift-ability, is the weight of the object, which the agent can feel during the lifting trial, e.g. if it is able to measure the force via a force sensor. But again this property can't be perceived directly. Instead hints like structure and colour allow to conclude about the lift-ability.

In general the physical properties that are responsible for a behaviour to cause a certain *outcome event* are most likely not to be visually detectable. Instead of learning these properties, the agent learns to find visual cues that are sufficient to anticipate different outcomes caused by these properties. One does not have to have a concept of weight, to be able to distinguish liftable from non-liftable objects.

This means that the agent does not gain a sophisticated concept of the properties (like physical laws) that are causing and influencing certain behaviour outcomes, but it can get a concept of what happens in its world if it applies behaviours on certain objects or parts of the environment.

2.2 Multi-Relational Affordance Knowledge

In the MACS project, affordances within a robotic system are represented by *cue event - behaviour - outcome event* relations. The cue events and outcome events, their interrelations as well as the relation to causing behaviours are learnt from the incoming perceptual data stream, using the learning approach that is described in deliverable D5.3.1. This knowledge is stored in a repository. For one instance of these relations the term *triple* is used.

This section shall emphasise that the data space does not consist of triples but that triples are derived from the learnt multi-relational repository. To be able to extract these triples is not only crucial for learning by self-experience and for planning but also for learning by imitation to match observed *cue events* and *outcomes* to previously made self-experience. That means that triples, which are 1:1:1-relations, are derived from that o:m:n-relatios database. Why the affordance repository is multi-relational is described in the following subsections.

2.2.1 One Behaviour Causing Multiple Outcome Events

Within the learning process, the *outcome event characteristics* are derived from the partitions of the behaviour related *application spaces* (see deliverable D5.3.1). This process could extract more than one characteristic for each of these partitions, since an application of a behaviour can cause multiple *outcome events*, e.g. pushing a light box leads to a displacement of the box, as well as to a change of the robots position and leads to free space where the box stood before.

One behaviour applied several times could also cause different *outcome events* depending on the configuration of the environment, e.g. the behaviour *stretch arm* can overthrow a box that is placed in front of the robot and in another case can cause a button to be pushed.

How characteristics are derived from experienced *outcome events* is influenced by what is relevant for the robot and for its mission as well as dependent on its level of knowledge. Therefore it is also possible, that multiple characteristics for one part of the *outcome event* are derived. For example if the robot kicks a ball, the structure of the trajectory can be characterised but also a second characteristic can be derived that describes that the ball has changed its location while neglecting the particular trajectory. Therefore, based on the level of abstraction and the focus on relevant properties, multiple different characteristics can be derived from one observation.

Because of the described facts the learnt *behaviour* - *outcome event* relations are $1 : n$.

2.2.2 Multiple Behaviours Causing Similar Outcome Events

For affordances like *move-ability* of an object or *traverse-ability* of a room, it is clear that they can be used by different behaviours, e.g. a room can be crossed by foot or by crawling. This means that several behaviours applied to the same environment cause the same *outcome event* "the room is crossed".

Figure 2 shows that the two behaviours *rapidly drive forward* and *rapidly stretch out arm* applied to the same environment (ball in front of the agent) can lead to the same *outcome events* (a straight moving ball).

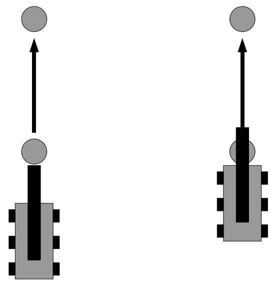

Figure 2: Two behaviours applied to the same environment can lead to the same *outcome event*.

But even if different *outcome events* occur by applying different behaviours, the characteristics of the observations could be equivalent, based on the level of interpretation as described in section 2.2.1. Figure 3 shows an example of how two different behaviours (*kicking a ball* vs. *beating a ball*) can lead to different *outcome events* and to same *outcome events* at the same time. Despite of different trajectories both outcomes have in common that the location of the ball has changed. Thus these *outcome events* could e.g. be characterised in one case by $\{c_1^o, c_2^o]\}$ and in the other case by $\{c_3^o, c_2^o\}$ whereby c_1^o and c_3^o are characteristics of the ball trajectories and c_2^o describes the displacement of the ball, which is equivalent for both *outcome events*.

In these cases the learnt *behaviour* - *outcome event* relations are $m : 1$

2.2.3 One Cue Event Indicating Multiple Behaviour Application Possibilities

During the learning process, *cue event characteristics* are derived from the partitions of the behaviours *application spaces* (see deliverable D5.3.1). This process can extract more than one characteristic for each of these partitions, since a *cue event* could be an indicator for multiple affordances, i.e. can be used as an indicator for an behaviour application to result in a related *outcome event*. For example the size of a reachable structure can be a *cue event* for action *close hand* to result in the *outcome event* one would name *gripped* as well as for action *move forward* to result in an an *outcome event* that could be labelled *entity moved*. Thus, one *cue event* can be characteristic for the detection of multiple

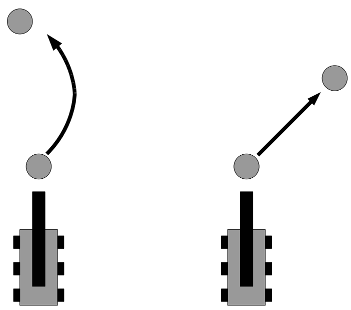

Figure 3: Two behaviours applied to the same environment can on the one hand lead to different *outcome events* that on the other hand share some properties. In this example the exact description of the balls trajectory differs, but in principle both actions cause a movement of the ball.

affordances.

In this case the learnt *cue event* - *behaviour* relations are $1 : m$

2.2.4 Multiple Cue Events Indicating One Behaviour Application Possibility

More than one *cue event* can be related to one and the same action and *outcome event* This means, that multiple *cue events* can be indicators to anticipate an *outcome event*, when a certain behaviour is applied. For example the size of an entity could be a *cue event* for the complex behaviour *lift* (consisting of the sequence of *close hand* and *lift arm*) to result in an *outcome event* that can be labelled *lifted*. Objects of a certain material however (e.g. polystyrene) could be liftable too, even if they are big. Thus another *cue event* describing the texture of the entity could be related to behaviour *lift* and outcome *lifted*. Therefore the learning process is able to result in more than one *cue event* to detect the possibility to apply one behaviour, and thus in a later stage to detect the existence of one and the same affordance.

In this case the learnt *cue event* - *behaviour* relations are $o : 1$

2.2.5 Multi-Relational Knowledge Base

To summarise the previous sections, the relation between *cue events*, behaviours, and *outcome events* is a $o : m : n$ relation and the members of a triple (c_1^c, b_1, c_1^o) can also be part of other triples, e.g. (c_1^c, b_3, c_2^o) etc. (see also figure 4). This means that the *affordance representation repository* is not only a database for triples, but a multi-relational database containing the *cue event characteristics*, behaviours, and *outcome event characteristics* and their relations.

This multi relational database allows the artificial agent to match observed events to *cue events* and *outcome events* learned by self-experience and thereby to gain knowledge about new interaction possibilities via observation and imitation as described in the following sections.

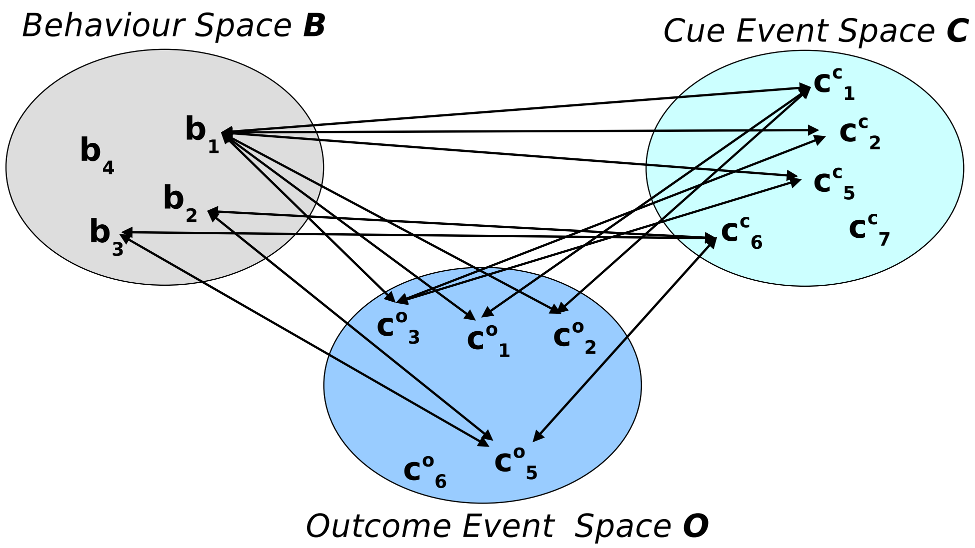

Figure 4: This figure shows that the result of the learning process are $l : n : m$ relations between behaviours, cue events and outcome events, which are characterised by c^o resp. c^c.

3 Building the Application Observation Space

The affordance learning architecture described in Deliverables D5.3.1, D5.3.2 and D5.3.3 enables the agent to learn what consequences the application of the agents behaviours have. These consequences are called *outcome events*. The *Outcome Event Space* $\boldsymbol{O}$ (see figure 4) contains descriptions of these outcome events that can later be used to monitor the application of a behaviour and decide whether the right outcome was achieved or not. These outcome event descriptions include descriptions about consequences within the agent's environment (external perception) and about changes of the agents state (proprioception and internal perception, including internal value systems).

The agent, continously observing its environment, is also able to perceive changes in the environment that are not caused by itself. An extension of the learning by self-experience architecture is, that the agent records this observations and tries to match them (also partially) to its own experience (to its stored outcome event descriptions). If there is a positive match the perceived data stream is to be stored for learning by observation. The space which is build by these observations is then called *Application Observation Space* $\boldsymbol{Ao}$ (see figure 5).

The process of matching observations to outcome event descriptions can be designed very simple by detecting only those events that exactly match the stored descriptions. This would make sense for example if the robot is guided to perform behaviours (that lead to outcomes it has already experienced) in a certain order to fulfil a task. This method would be time efficient, but would at the same time limit the number of possible learning scenarios. For example the robot Kurt3D[1], used in the MACS project, is equipped with a crane arm to be able to lift objects. This lifting is of course different from the way humans lift objects and thus an observation of a human lifting an object would for the robot most likely differ from the self-experienced lifting.

To avoid this limitation the matching process has to include similarity measurements and abstraction methods. These methods could for example include a matching of discrete states within the observation and the stored descriptions. This would be in line with the observations and theories documented done by Edwards in [EGC03] who showed, that in their experiments children imitated goals of actions instead of the actions themselves.

[1]http://www.ais.fraunhofer.de/ARC/kurt3D/

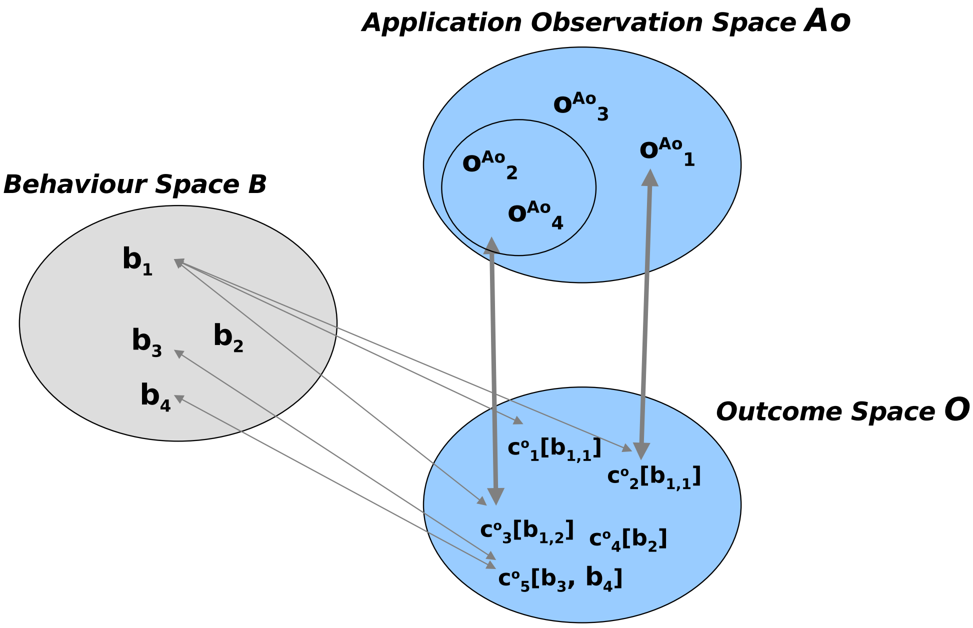

Figure 5: Matching observations to *outcome events* the agent gained by self-experience is the basis on which the agent can learn by observation and thus imitate.

4 Learning new Cues

Section 3 describes the recording of observations that match one or more of the *outcome events* the agent learned by self experience. For example the description of the *outcome event "something is lifted"* could contain the description of the ascending spatiotemporal trajectory of a colour blob. Thus the agent is enabled to observe lifting actions by observing ascending colour blobs in its environment.

During the self-experience learning process the agent builds up a database of *cue event* descriptions, which are indicators, that the agent can use to anticipate which *outcome event* will be caused by the application of a behaviour on or with the particular *cue event.*

When an observed inter-action includes entities the agent did not encounter before or entities on which the agent yet did not try to apply a behaviour, the agent is enabled to extract new *cue events* that are related to the observed *outcome event* and thus could be related to the *outcome events* the agent learned by self experience. In the example of a lifting experiment, the agent could observe a human lifting cans, pencils or mobile phones, while itself before only performed and thus self-experienced lifting of simple building blocks (see figure 1).

For this extraction of potential new *cue events*, respectively of *cue event descriptions* the learning steps 2 and 3 for channel-extraction and event-description (see deliverable D5.3.1) as developed and implemented in the architecture for learning affordances by self experience, can be re-used.

Affordances do not only depend on the physical properties of the environment, but are relations between the environment, its properties and an agent. Thus the newly derived potential *cue events* are to be treated as hypothesis by the agent, since the observed agent could have different physical properties, like stronger muscles, more flexible actuators, is bigger, or has actuators of a different quality (like a magnet in case of a robot). Those agents could for example lift entities that are heavier, larger, or of a different material than the entities, the agent can deal with.

The affordance based architecture has thus to include control mechanisms that force the agent to try to achieve a certain outcome by applying all actions that are related to the concerning *outcome event.* In case of achieving the previously observed *outcome event* (eventually multiple times) the new *cue even description* can be related to the action and the *outcome event.* The module storing the *Application Space*s thus has in addition to include a data base for potential *cue events.* (see figure 6)

Using these methods, the agent is enabled to learn to associate new *cue events* to the already known *outcome events*, including observation and imitation. In addition, the information about those *cue events* that could not be proven to be associated to the concerning *outcome event* could be used to gain knowledge about affordance, that other agent can use. Therefore the agent would have to be extended by a system that detects other agents or parts of these agents (see section 5.1).

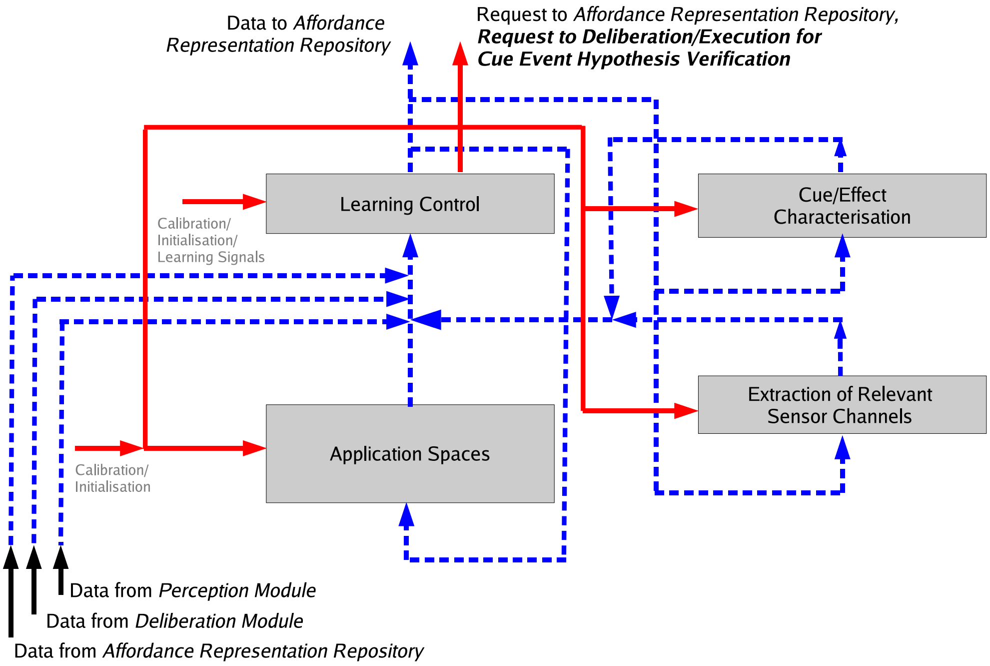

Figure 6: Modified affordance learning architecture (and reused modules) to support learning of cue events by observation and imitation. In addition to the architecture, the *Application Spaces* module has to include a storage system for potential *cue events* and the *Learning Control* must have the possibility to influence the agents deliberation and execution for hypothesis verification.

5 Learning new Outcomes

The methods described in this section are designed to enable an artificial agent, that is equipped with an affordance based control including the affordance learning architecture, described in Deliverables D5.3.1, D5.3.2 and D5.3.3, to extend its knowledge about the consequences (*outcome events*) of applying actions to the environment, by observing the consequences of the application of behaviours by other agents.

The learning of new *outcome events* by observation and imitation enables the artificial agent to subsequently apply the learning of new *cue events* by observation (see section 4).

5.1 Learning Outcomes via Behaviour Cues

The *Application Observation Space* introduced in section 3 contains data that was recorded during (an slightly before) the observation of a known *outcome event*. In case these trajectories include *cue events*(s) already known to the agent (because known entities are involved), the system can relate the observation to the observed *outcome event* and to the *cue event*(s). Thus the observation can be related to one or more behaviours (in the *Behaviour Space* of the agent), that are related to the concerning *outcome event* and *cue event*(s).

When the agent observes an *outcome event*, the observation should in most cases include information about the causing actor. The *cue event* extraction mechanisms provided by the architecture for learning affordances by self experience can be re-used to extract *cue event description*(s) (c_i^b) for the detection of the acting agent. Since the *Application Observation Space* is partitioned into sets of observations related to the same behaviour (in *Behaviour Space* $\boldsymbol{B}$) the application of the *cue event description* extraction methods to these partitions could lead to *cue event description* that can be used by the system to anticipate a certain action of the observed acting agent.

The *cue events* for the detection of behaviours form the *Behaviour Event Space* $\boldsymbol{BE}$ (see figure 7). These cues can now be used to detect an acting agent or the application of a behaviour, even if the caused *outcome event* is unknown to the agent. Thus these observed *outcome events* can be extracted and included into the agent's data base. In addition to this extraction, the agent has to extract new *cue events* related to the *outcome event* and the causing action.

However as in the case of learning new *cue events* in section 4, the *outcome events* and the related *cue events* learned by observation of *behaviour cue events* must be treated as hypothesis, since the other acting agent could have differing physical properties (size, strength, etc., see above). Thus a process is needed that verifies, if the observed *outcome events* and *cue events* are also valid for the agent. This process is not limited to a true/false decision, but could also include a process of deriving new *outcome events* in case, that the trials resulted in “nearly” the expected outcome or fulfils just a part of the description..

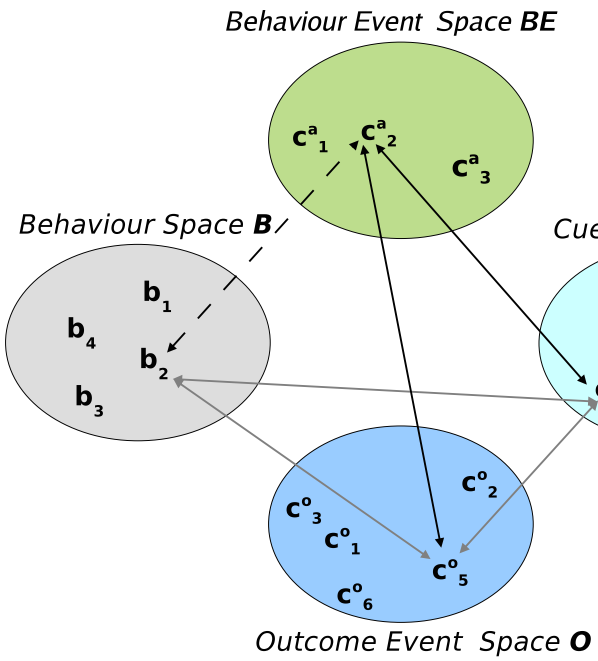

Figure 7: When the agent observes an *outcome event*, the observation should in most cases include information about the causing actor. The *cue event* extraction mechanisms provided by the architecture for learning affordances by self experience can thus be re-used to extract cues for the acting agent. The *cue events* for the detection of behaviours (c_i^b) form the *Behaviour Event Space* ***BE***.

5.2 Learning about Outcomes through Observation of Action Sequences

5.2.1 Learning to Cause New Outcome Qualities

The *outcome events* learned by the architecture for learning affordances by self-experience are designed to describe the outcome of the application of a behaviour in an abstract way. For example the agent could have learned, that it can stack a building block on top of another building block. This description only contains information about the fact that one building block is on top of the other at the end of the application of the related behaviour. It does not tell anything about the quality of the stacking, e.g. how stable it is and how many other building blocks one can stack on top of it.

Observing another actor performing a sequence of stacking behaviours, could enable the agent to gain knowledge about the sequence in which objects are to be stacked to build a tower as high as possible. This qualitative knowledge could be added to the *outcome event characterisation*, e.g. as parameters that can be optimised by the deliberative part of the overall affordance based control architecture into which the affordance learning architecture is embedded. Thus the agent can be trained by a teacher to build very high towers or to kick a ball very hard to shoot a goal.

5.2.2 Extending the Affordance Sequencing Possibilities

The usage of sequences of action possibilities (affordances) leads to a sequence of outcome events that can be detected detached from each other, e.g. after using the push-ability of a box in the middle of a hallway, the achieved traverse-ability of the hallway can be used. These basic affordances like push-ability or traverse-ability can be learned by the basic learning approach described in deliverable D5.3.1 (which is realised in the described learning architecture). To derive the outcome of such sequences presumes that the outcome of a behaviour somehow encodes the cues that are related to the next desired outcome via the successive action. This might not be the case for all outcomes.

To solve this problem, two new types of relations are introduced in addition to the relations between behaviours, cue events, and outcome events:

- Relations between behaviours.
  These relations are strengthened, when the actions are used in sequence, and decay when one of the actions is used without the other.

- Relations between outcome event characteristics and cue event characteristics.
  These relations are strengthened, when a cue event occurs after the outcome event was detected, and decay, if the cue event is not detected sub-sequent to the outcome event.

The emerging relations (see figure 8) can be used to find paths that lead from one perceived affordance to another desired affordance or to a desired outcome event, if the outcome event description itself does not contain the necessary next cue event.

The relations between the outcome event characteristics and cue event characteristics can also be build by observing other acting agents. Therefore the agent has to monitor its environment and detect outcome events in its environment without acting himself. This enables the agent to learn to use sequences of affordances by observation.

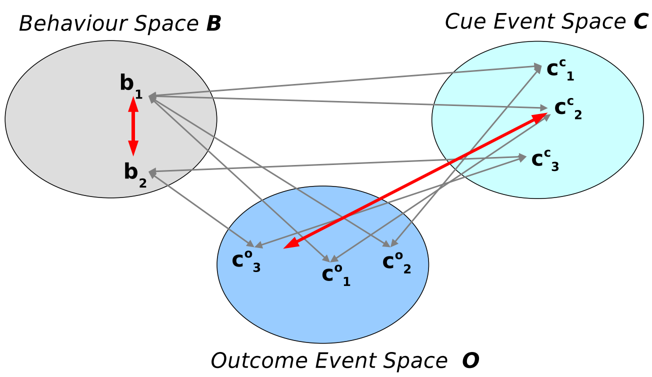

Figure 8: Relation between behaviours and between outcome event characteristics and cue event characteristics could be established to enable the robot to find possible action sequences that lead from one perceived affordance to another desired affordance or to a desired outcome event.

5.2.3 Observing Behaviour Cues for Learning new Outcomes of Action Sequences

When a sequence of actions is performed by the agent, an outcome can emerge that can not be anticipated by sequencing the application of multiple actions and their related outcomes. A simple example is lift-ability: The behaviours "close hand" and "lift arm" do lead to a different outcome if used detached from each other, than in case of using them in sequence to lift something. The process of learning affordances by self experience described in Deliverables D5.3.1, D5.3.2, and D5.3.3 therefore incorporates the detection of action sequences and components for learning complex actions that are composed out of sequences of simple actions via trial and error.

However the search space is growing rapidly with the length of the action sequence. This learning process can be accelerated if the agent uses the ability to observe behaviour applications on the basis of behaviour cues, as described in section 5.1. The sensor data perceived during the time the observed action is performed, is to be recorded for this learning process. As in the case of the learning process for basic affordances (Deliverable D5.3.1), the data, recorded during the observation should include a certain amount of data of the pre-application and of the post-application phase.

The same learning mechanisms that are used within the affordance learning architecture described in the Deliverables 5.3.1, D5.3.2, and D5.3.3 can then be used to extract cues and outcomes out of the the recored time series.

This is in accordance with the ideas developed and documented in Deliverable D5.1.1 and follows the ideas described in [CH03]:

> This paper does not deal with affordances explicitly, but it provides a method which can be adapted to learning affordances by observation and is thus relevant for MACSs work package 5.4. Preconditions of an action include affordances, e.g. when an object does not afford gripping the action grip cant be performed. When an agent has learned a set of basic affordances, it can observe other artefacts and map perceived actions to its own action possibilities using the perceived states and affordances. The resulting chains of actions together with the necessary pre- and post-conditions provide new complex actions which result in new more complex affordances. E.g. a robot can learn by imitation to grip object A, lift it, move it to another location, and put it on top of object B; this chain of actions could be called piling and the objects that are used afford piling (which could be seen as a compilation of the affordances of gripping, lifting, moving, lowering, unhanding of object A and of support of object B). This relieves one from hard coding all possible actions and thus predefining the set of affordances that can be learned by the artefact.[DIKP04, p.6]

The agent is thus enabled to learn new outcomes (and related cues) of action sequences by observation without necessarily interacting with its environment itself. Learning thus does not only have an individual dimension, but becomes a social function.

6 Summary

In the previous sections, it was shown how the architecture for learning affordances by self experience, described in Deliverables D5.3.1, D5.3.2 and D5.3.3, is extendet to enable an artificial agent to enhance its knowledge about affordance usage by observation and imitation. This does take place by using the introduced extensions:

- Learning new *cue events* by observing known *outcome events*
- Learning new *outcome events* by using *behaviour cue events*
- Learning to use sequences of affordances through observation
- Learning new affordances through the observation of *behaviour cue events*

Using these mechanisms, the system is able to learn affordances (or sequences of using affordances in a certain order) limited by the agent's capability space which is defined by the behaviours the agent is able to perform. Furthermore it is possible that the agent gains knowledge about affordance usages without beeing able to imitate them by extracting knowledge about the physical properties of other acting agents from the observations of outcomes, as explained in section 4 and 5.1.

One has to regard, that for all the proposed extensions of the architecture to include imitation and learning by observation, the agent does not have to "be aware" of another acting agent. Nevertheless the system could enable a population of agents to increase their learning speed by observation and imitation even though the individual agents have no explicit knowledge about other agents.

References

[CH03] Carlos Antonio Acosta Calderon and Huosheng Hu. Goal and actions: Learning by imitation. In *Proceedings of the Second International Symposium on Imitation in Animals and Artifacts*, 2003.

[DIK06] Georg Dorffner, Jörg Irran, and Florian Kintzler. Robotic learning architecture capable of autonomously segment action sequences into affordances. Deliverable MACS/5/3.2 v1, österreichische Studiengesellschaft für Kybernetik (öSGK), Vienna, Austria, 2006.

[DIK07] Georg Dorffner, Jörg Irran, and Florian Kintzler. Robot prototype learning affordances through self-experience v2.0. Deliverable MACS/5/3.3 V2, österreichische Studiengesellschaft für Kybernetik (öSGK), Vienna, Austria, 2007.

[DIKP04] Georg Dorffner, Jörg Irran, Florian Kintzler, and Patrick Pölz. Overview of existing affordance learning approaches. Deliverable MACS/5/1.1, Österreichische Studiengesellschaft für Kybernetik (ÖSGK), Vienna, Austria, 2004.

[DIKP05] Georg Dorffner, Jörg Irran, Florian Kintzler, and Patrick Pölz. Robotic learning architecture that can be taught by manually putting the robot through action sequences. Deliverable MACS/5/3.1 v1, Österreichische Studiengesellschaft für Kybernetik (ÖSGK), Vienna, Austria, 2005.

[EGC03] Martin G. Edwards, Glyn W. Humphreys, and Umberto Castiello. Motor facilitation following action observation: a behavioural study in prehensile action. In *Brain Cognition*, volume 53, pages 495–502, 2003.