ExpCog: Experiments in Commonsense Cognitive Robotics

Manfred Eppe[1], Mehul Bhatt, Jakob Suchan, Brian Tietzen[2]
www.commonsenserobotics.org

Abstract. ExpCog is a high-level cognitive robotics framework aimed at integrating logic-based and cognitively-driven agent-control approaches, qualitative models of space, and the ability to apply these in the form of planning, explanation and simulation in a wide-range of robot-control platforms and simulation environments. In addition to its primary experimental function, the framework also has a utility toward didactic purposes, e.g., as a teaching and experimental aid in courses on Artificial Intelligence, Knowledge Representation and Reasoning, and Robotics.
We demonstrate the ExpCog framework in the backdrop of an online incremental planning and postdiction driven autonomous wheelchair robot control task. We propose that ExpCog, if pursued as a long-term initiative in an open-source format, promises to consolidate KR methods for robotics under a unifying platform, thereby offering researchers, students, and enthusiasts alike direct access to commonsense cognitive robotics.

INTRODUCTION

Commonsense reasoning about physical activity grounded action, control, and interaction promises to become a fundamental aspect of robotic assistance systems and technologies that will accompany us in everyday personal and professional activities. Formal representational and computational methods for handling commonsense qualitative phenomena – e.g., involving the perception and reasoning about space, time, events, actions, change, causality, processes - will be at the heart of collaborative cognitive systems and assistive technologies aimed at high-level control of autonomous robotic systems with common sense. Methods and tools for integrated commonsense reasoning about space, actions and change are therefore of paradigmatic importance from the viewpoint of both theoretical as well as application-driven goals [3].
There exists a variety of different knowledge representation (KR) formalisms, as well as a variety of different robotic platforms, and this results in difficulties regarding the seamless interchangeability of different reasoning engines for different robots. Researchers, students, and enthusiasts who are interested in the field of (commonsense cognitive) robotics require the ability to easily experiment with different agent modelling and control approaches without having to deal with the complexity of the underlying technical apparatus (e.g., complex hardware setup). Additionally it should be possible to compare different control approaches, logical based as well as cognitive driven within the same domain, and to quick-start the software engineering of robotic systems. Especially, students desirous of learning about high-level control approaches should be able to build upon integrated low-level actions for a range of robots, thereby allowing them to focus their learning to the high-level control program.
To this end, we propose *ExpCog* – a framework that integrates robotic platforms and reasoning engines in a modular manner, such that the individual components can be plugged in and out freely, without much effort for the system designer. We do not claim to have solved every problem that inheres the construction of such a framework, but, as we demonstrate in this paper, we achieved the core functionality to combine different reasoners with different robotic platforms. The development of ExpCog emphasises the practical integration and application within large-scale robot control frameworks (e.g., ROS[3], iCub[4], v-rep[5]), as well as toy robots (e.g., LegoMindstorms[6]) and light-weight simulation scenarios.
In this paper, we demonstrate the capabilities of the ExpCog framework in the context of a real-world deployment in the Bremen Ambient Assisted Living Lab (BAALL) [24], which consists of the Rolland autonomous wheelchair situated within a smart home together with a range of other sensing and actuating technologies. The presented autonomous wheelchair control scenario involves online incremental planning, sensing, and abductive reasoning aimed at making sense of incomplete and conflicting narrative knowledge in the presence of abnormalities. In the demo scenario, the online planning and sensing is realised using the situation calculus based *Indigolog* programming language [12], and abductive reasoning has been modularly integrated based on the *h-approximation* ($\mathcal{HPX}$) method [15], a recently developed epistemic action theory tailored towards efficient postdictive and abductive reasoning. The demonstrated scenario illustrates the versatility of the ExpCog framework in modularly integrating two distinct reasoning mechanisms, and their seamless deployment in controlling a state of the art real world robot.

RELATED WORK

Commonsense Cognitive Robotics

Indeed, there have been many advancements in the theoretical foundations of commonsense reasoning about space, actions, and change, and the development of tool-sets in the respective contexts (see e.g. [50]).

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Spain
email: meppe@iiia.csic.es
[2] University of Bremen, Germany
email: {bhatt,jsuchan,btietzen}@informatik.uni-bremen.de

[3] Robot Operating System (ROS). www.ros.org
[4] iCub: Open Source Cognitive Humanoid Robotic Platform. www.icub.org
[5] Virtual Robot Experimentation Platform (v-rep). www.coppeliarobotics.com
[6] Lego Mindstorms Platform (EV3RSTORM). http://mindstorms.lego.com

However, the delivery of basic outcomes from these research communities into the hands of practitioners of robotics is still an ongoing challenge, in particular with respect to research that would emphasise and demonstrate the synergies resulting from integrating high-level commonsense reasoning at the symbolic level with low-level statistical and quantitatively driven methods within robotics [48], e.g., concerning vision, localisation, motion control.

Recent efforts within the *RACE* project[7] [41, 33] try to incorporate a multitude of AI skills like learning, planning and other high-level reasoning tasks for the application in a service robot. The combination of such reasoning tools results in exiting new possibilities, but the *interchangeability* of different robotic platforms and reasoning tools seems to be a minor issue in the project.

A promising approach that allows robotic platforms to access different knowledge bases is the *KnowRob* framework [45]. The focus of the framework is on inferring and accessing ontological knowledge that is required to execute vaguely specified tasks on-the-fly. The approach has been proven successful for various object manipulation tasks such as cooking, but it has less experimental character than ExpCog. For example, its causal reasoning and planning ontology [44] is fixed, which makes it hard to experiment with different action calculi, or with spatial calculi like RCC [8] and $\mathcal{OPRA}$ [13].

The framework presented in [25] is similar in that it constitutes a knowledge architecture for robots. However, its focus is more on cognitively motivated human-robot interaction and action/plan learning, with little emphasizes on the experimental character of robotic frameworks. The *Aquila* framework [38] is another example for a cognitive robotics software architecture, but heterogenous support for different reasoning tools is not provided. This makes it difficult to perform experiments with different reasoners. Instead, the framework focuses more optimizing access to the hardware that is required for the reasoning (e.g. GPU processing).

Action and Change

Research in the field of Reasoning about Actions and Change (RAC), also referred to as Cognitive Robotics [27], has considerably matured. Over the last decade, some of the theoretical work and the resulting formalisms for representing and reasoning about dynamic domains have evolved into practically applicable high-level agent control languages, some examples here being the situation calculus based GOLOG [28] family of languages, e.g., CCGOLOG [22], CONGOLOG [10], INDIGOLOG [12], which is an incremental deterministic version of CONGOLOG, and the fluent calculus based language FLUX [47]. Differences in the theoretical underpinnings notwithstanding, a common feature of all these languages is the availability of imperative programming style constructs for the domain of robotics/agent-control, i.e., statement in the program correspond to actions, events and properties of the world in which an agent is operating. There have also been considerable advances in the standardization and benchmarking of domain description languages in the form of the Planning Domain Definition Language (PDDL) [35] and related initiatives.

A popular means for the formalization of action and change is answer set programming (ASP) [20]. ASP provides a truly declarative, nonmonotonic and efficient logic programming interface. Among the various action languages have been implemented in ASP (e.g. [31, 26, 49]), we are particularly interested in the *h-approximation* ($\mathcal{HPX}$) approach [15], because it efficiently supports incomplete knowledge and sensing actions, along with different forms of abductive reasoning that we employ for abnormality detection.

[7] http://project-race.eu/

Spatial Representation and Reasoning

The field of Qualitative Spatial Reasoning (QSR) has emerged as a sub-division in its own right within KR [9, 40, 1, 30, 2]. Research in QSR has focused on the construction of formal methods (i.e., qualitative spatial calculi) for spatial modelling and reasoning. The scope of QSR, at least in so far as the context of qualitative spatial calculi is concerned, has been restricted to representational modes for spatial abstraction and reasoning. Major developments in this regard include: (a) the development of spatial calculi that are representative of distinct spatial domains, (b) constraint-based techniques for ensuring the global consistency of spatial information and (c) development of general tools for spatial abstraction and reasoning.

THE EXPCOG FRAMEWORK

Overview & Architecture

ExpCog presently integrates diverse logic-based action and change / control calculi, qualitative models of space, and the ability to apply these in the form of planning and explanation with a wide-range of robotic-control platforms and simulation environments. The objectives that guide the framework design and implementation are:

O1 It should be possible for a domain-modeller to specify the domain theory and commonsense dynamics and exploit more than one control approach thereafter, without the need to dwell on the details of any of the available control approaches, or having to re-model the domain with respect to the requirements of a particular action language.

O2 The framework provides easy integration with existing low-level control apparatus such as robot control and simulation interfaces that exist in the open-source and commercial domain, some example of these being ROS, Gazebo, V-Rep, iCub etc.

O3 The framework supports easy experimentation with different control techniques also combining multiple control techniques, and provides general modes of spatial information representation and reasoning, and additionally seamlessly integrates new low-level robot control apparatus.

O4 In addition to serving its primary experimental function, it is also useful for teaching purposes in courses on AI, KR, and robotics.

Figure 1 presents a conceptual overview of the ExpCog architecture; the main aspects are elaborated on in the following in (C1–C3):

C1. Action Reasoning and Control

The primary aim of the framework is to provide a suite of different control approaches that may be used for representing and reasoning about dynamic environments. The suite of control approaches available with the framework also constitutes the most important (functional) component of the overall experimental framework. It consists of a collection of several different formal techniques, both logic-based and cognitively-driven models. These can be used as control mechanisms in robotic domains, or to reason about changing spatial environments in general. Control approaches based on the following formalisms will be available for use in an *independent* manner within the proposed framework:

- a basic STRIPS like planning system [16] and Belief-Desire-Intention (BDI) approach [7] implemented in SWI-Prolog
- event calculus [23] based DEC reasoner [37]
- fluent calculus [46] based FLUX language

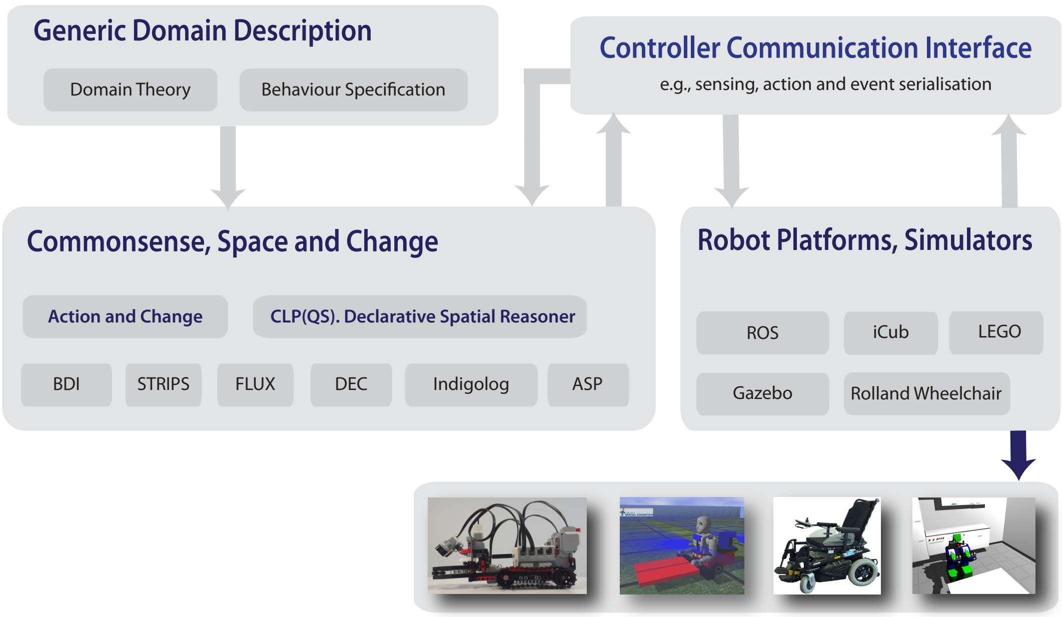


Figure 1. ExpCog – Conceptual Overview of Architecture

- situation calculus [34] based languages Golog [28] and Indigolog [12]
- answer set programming based *h-approximation* ($\mathcal{HPX}$) (driven by oclingo) [15]

The inclusion of additional reasoning and control methods is a topic of ongoing work, and is expected to be primarily driven by open-source community contributions in the future.

C2. Commonsense Spatial Reasoning

Central to the long-term development vision of ExpCog is the notion of a domain-independent *qualitative spatial theory*, which is representative of an underlying 'qualitative physics' that is applicable for a wide-range of *dynamic spatial systems* [5]. Here, a dynamic spatial system refers to a specialization of the dynamic systems [42] concept for the case where a domain theory consists of changing qualitative spatial relationships pertaining to arbitrary spatial aspects such as the orientation [18, 36], direction [17, 29], and topology [39]. Basically, what this implies is that qualitative spatial relationships are modelled as fluents within the domain theory of a robot control program, and the semantics for the spatial relationships is offered by a generic spatial representation and reasoning engine (namely CLP(QS)) that is embedded within the ExpCog framework.

CLP(QS): A Declarative Spatial Reasoner CLP(QS) is a declarative spatial reasoning system capable of modelling and reasoning about qualitative spatial relations pertaining to multiple spatial domains, i.e., one or more aspects of space such as topology, and intrinsic and extrinsic orientation, distance [4, 43].[8] Furthermore, users / application developers may freely mix object domains (i.e., points, line-segments, and regions) and with the available spatial domains. CLP(QS) also offers mixed geometric-qualitative spatial reasoning capabilities, and in its current form, a limited support for quantification offers the means to go back from qualitative relations to the domain of precise quantitative information. CLP(QS) is implemented as a general library within the context of Constraint Logic Programming (CLP). CLP(QS) may be used in robotics projects via the ExpCog framework, or it may be used independently via a logic programming based interface (i.e., Prolog-based library) within a range of large-scale cognitive systems and assistive technologies.

The notion of a domain-independent qualitative spatial theory within the framework is primarily used as a means to demonstrate the applicability of (existing) qualitative spatial models relevant to different aspects of space in realistic dynamic spatial scenarios in robotics. In addition, such a theory has the advantage of being general and re-usable in a dynamic spatial domains outside of the purview of robotics (e.g., cognitive vision, geospatial dynamics)

C3. Robot Platform Independence & Extension

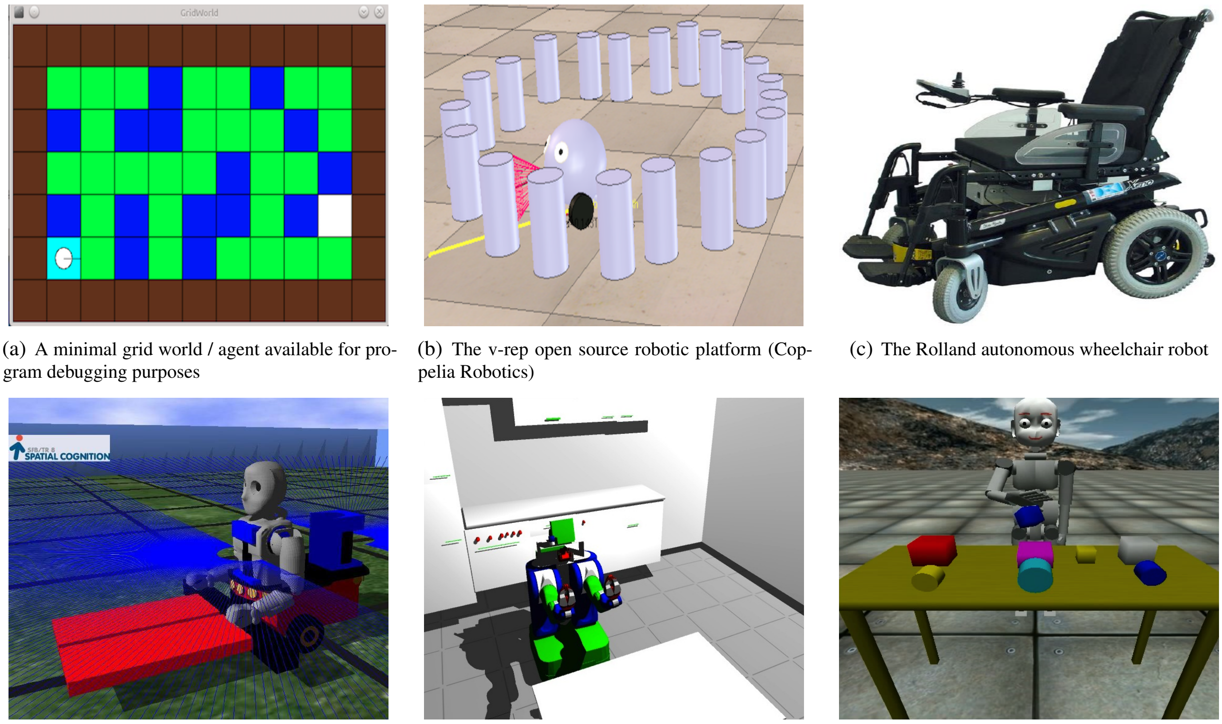
The ExpCog framework is independent of any particular robotic system/platform or agent simulation environment, thereby ensuring applicability in a wide-range of real or simulated environments. For this purpose, an adequate level of abstraction between the experimental framework and robotic hardware and simulated systems have been implemented.

Controller Communication Interface (CCI). ExpCog consists of a Controller Communication Interface (CCI) that provides the necessary abstraction between robotic or simulation platforms and high-level knowledge representation and reasoning modules. This independence is achieved by the generic CCI by explicitly defining all possible modes of communication (e.g., by way of serializing control actions to the robot's actuators and the inflow of sensing information) between the framework and the external world (real robot, or simulator) that is being interfaced with.[9]

Supported Platforms. Figure 2 illustrates the robot platforms that have been integrated within ExpCog. Sample scenarios involving the

[8] CLP(QS): `www.spatial-reasoning.com`

[9] The CCI layer is implemented such that the the high-level reasoning and low-level action control may be executed even in distributed locations over a network connection.



(a) A minimal grid world / agent available for program debugging purposes

(b) The v-rep open source robotic platform (Coppelia Robotics)

(c) The Rolland autonomous wheelchair robot

(d) A Pioneer-driven delivery robot based in the Gazebo platform equipped with a gripper, and SICK laser

(e) ROS based PR2 robot platform in a kitchen domain

(f) The iCub robot platform

Figure 2. ExpCog Supported Robot Platforms and Demo Scenarios

supported robot platforms have been included such that they may be used as tutorials (of high-level representation and reasoning). We also emphasise that it is possible to integrate new robotic platforms within ExpCog relatively easily in a modular manner by writing new device controller(s).

EVLUATION AND DEMONSTRATION

An Autonomous Wheelchair Robot in a Smart Home

In order to give the reader an idea of the required effort to use ExpCog, and also as a practical evaluation and demonstration, we present a scenario and the implementation of the underlying reasoning mechanisms. The scenario takes place in the Bremen Ambient Assisted Living Lab (BAALL) [24] (Fig. 4), a smart home environment with diverse actuators and sensors to realize situated and context-dependent activity assistance, especially for elderly and physically disabled individuals. Amongst its actuators and sensors are automatic doors, a movable kitchenette, a smart fridge and dynamic illumination control. BAALL also consists of the autonomous robotic wheelchair Rolland [32] (Fig. 2(c)) that is tailored towards situated user assistance in smart environments. One of the basic features of Rolland is the capability to autonomously navigate in narrow spaces, which is realised an independently executing low-level Simultaneous Localization and Mapping (SLAM) ([48]) module, and a route graph abstracted waypoint based navigation system.

Supporting Efficient Planning with Abductive Narrative Interpretation

The ongoing scenario requires narrative-based incremental agent control based on *plan monitoring*, and combining it with mechanisms for *abductive reasoning*. For example, if a monitored world property changed unexpectedly, the system abduces possible explanations that describe what may have happened that caused this change.
In the case of this scenario, ExpCog employs Indigolog's very efficient prioritized interrupt architecture [21] for online action planning. However, Indigolog's efficiency comes at the cost that its epistemic reasoning capabilities are limited [11]; in particular, Indigolog is not capable of two forms of abductive reasoning that are required for the scenario, namely (i) *postdictive reasoning* and (ii) *abductive explanation*.

(i) *Postdictive reasoning* is a crucial inference mechanism for abnormality detection in robotic environments. It allows one to diagnose why an action was (un)successful, and thereby helps to make robotic environments more error-prone. In general, postdiction can be understood as the inference about the condition of an action by observing its effect. For example, in the scenario we have a drive action that causes a the wheelchair robot to arrive at its destination under the condition that there is no abnormality that prevents the driving from being successful. If the agent observes, that the driving was not successful, then it can postdict that there must have been an abnormality that caused the action to fail.

(ii) *Abductive explanation* is the inference about an unexpected exogenous action that explains an unexpected world change. For example, if the system monitors that the door to the bathroom was closed unexpectedly, then it can explain the closed door with the exogenous action of a human closing the door.[10]

[10] Note that this explanation would trigger the additional postdictive inference that the person closing the door must be standing next to the door. Note also, that abductive explanation can only propose candidate explanations.

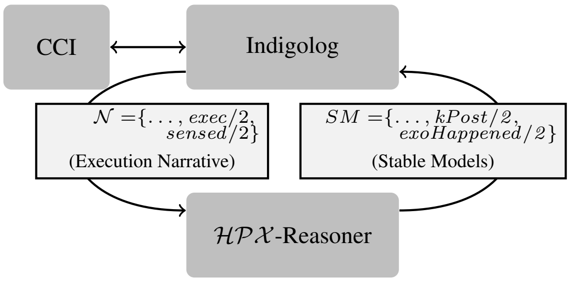


Figure 3. Information flow between Indigolog and the ASP-based $\mathcal{HPX}$ reasoner

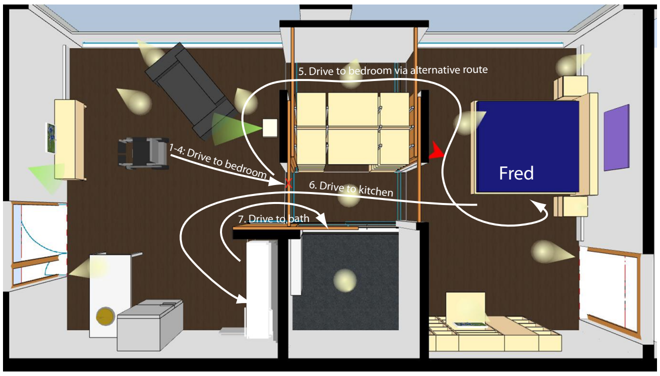


Figure 4. Schematic simulator view of the BAALL Smart Home Scenario

In order to realize these reasoning tasks, we employ the *h-approximation* ($\mathcal{HPX}$) formalism [15] and combine it with Indigolog in a supportive manner. $\mathcal{HPX}$ is less efficient than Indigolog in terms of action planning, but it has better sensemaking capabilities in that it is capable of performing both postdiction and abductive explanation. We depict the information flow between Indigolog and $\mathcal{HPX}$ in Figure 3.

Indigolog generates a plan and controls the wheelchair robot via ExpCog's Controller Communication Interface. The sensor data obtained during plan execution is parsed and then sent back to Indigolog, which in turn informs the $\mathcal{HPX}$ reasoner about the executed narrative (denoted $\mathcal{N}$) in an online manner. This is realized by sending $exec/2$ and $sensed/2$ atoms to the underlying online ASP solver *oclingo* [19] whenever an action was executed, resp. when a sensing result was received. This allows *oclingo* to continuously (re-)interpret the execution narrative, and to enrich it with postdicted and abduced information. The stable models that result from the online ASP reasoning are parsed and the gained information is sent back to Indigolog. This happens via (i) $kPost/2$ atoms that represent postdicted information, and (ii) via $exoHappens/2$ atoms that represent the occurrence of exogenous actions that where inferred with $\mathcal{HPX}$'s abductive explanation capabilities. Both kinds of information are translated into the occurrence of exogenous actions within Indigolog, so that Indigolog can react accordingly using its prioritized interrupt mechanism. A detailed trace of the communication between Indigolog and $\mathcal{HPX}$ for the particular use case presented in this paper is depicted in Fig. 5.

For example, it may also be possible that the door closed due to a wind breeze. Selecting the best candidate explanation is subject to future work.

Scenario Description

The basic scenario takes place in the BAALL; a schematic view is provided in Fig. 4.

1. The Rolland wheelchair is in the living room at the charging station
2. The person "Fred" is in the bedroom, and requires the assistance of the wheelchair in order to go to the kitchen; the user issues a command via a dialog based interface (e.g., "*take me to the kitchen*")
3. The wheelchair autonomously navigates toward the bedroom; during motion, control transfers to the low-level SLAM module thereby necessitating execution monitoring and control by the high level reasoning module
4. An abnormality is detected: the wheelchair cannot move because of an obstacle, and hence must re-plan
5. The wheelchair follows an alternative plan to reach the bedroom, and reaches its target location via a different route
6. Fred gets onto the wheelchair and drives to the kitchen. The kitchenette is lowered automatically, so he can reach it from the wheelchair
7. Thereafter, Fred needs to drive to the bathroom. While he is driving there, his wife Mary enters the bathroom and closes the door. Since Mary is the only other person in the BAALL, the system will correctly explain the closed door by abducing that that Mary has closed the door. This information can be used e.g. to issue a gentle voice command to Mary, explaining that Fred also wants to get to the bathroom

With respect to this scenario, we illustrate: (a) the core aspects concerned with modelling the overall knowledge exchange architecture involving integrating planning and postdiction; and (b) demonstrating select parts of the domain model for reasoning about action and change using Indigolog and Answer Set Programming.

Interoperation of Indigolog and $\mathcal{HPX}$ in BAALL

During plan execution, narrative-based knowledge about the environment is obtained via the ExpCog framework and transferred to the $\mathcal{HPX}$ module. The Indigolog controller is used as main decision making module that provides the execution narrative $\mathcal{N}$. The ASP-based $\mathcal{HPX}$ reasoner interprets the execution narrative and generates stable models (SM) of a logic program that consists of domain-specific background knowledge $\mathcal{D}$ and the execution narrative $\mathcal{N}$.

The domain specification of Indigolog can be mapped directly to $\mathcal{HPX}$, since it considers the very similar language elements, such as causal laws and initial knowledge definitions. For example, an Indigolog statement `causes_val(`a`,`f`,`v`,`c`)`, saying that an action a causes fluent f to have value v if condition c is true, can be translated to $\mathcal{HPX}$'s PDDL-like input syntax `(:action` a `:effect when` c `then` f`)` (assuming boolean fluents and v = `true`).

In the depicted scenario, the wheelchair first tries to drive autonomously to the bedroom. However, after executing the `senseloc(rolland,bed)` action it observes that it did not arrive at the bed. This information is sent to $\mathcal{HPX}$, that in turn sends the information about an abnormality `ab_drive(liv,bed)`. The wheelchair is now replanning and taking the route through the upper corridor `corr1`. On that route, no abnormalities are encountered and the wheelchair arrives at the bed, where Fred sits down on the wheelchair and drives into the kitchen. When approaching the kitchen, the kitchenette is automatically lowered so that Fred can access it.

After working in the kitchen, Fred drives the the bathroom. Here, the system monitors that the bathroom door is closed and explains this fact with the exogenous action of Fred's wife Mary closing the door.

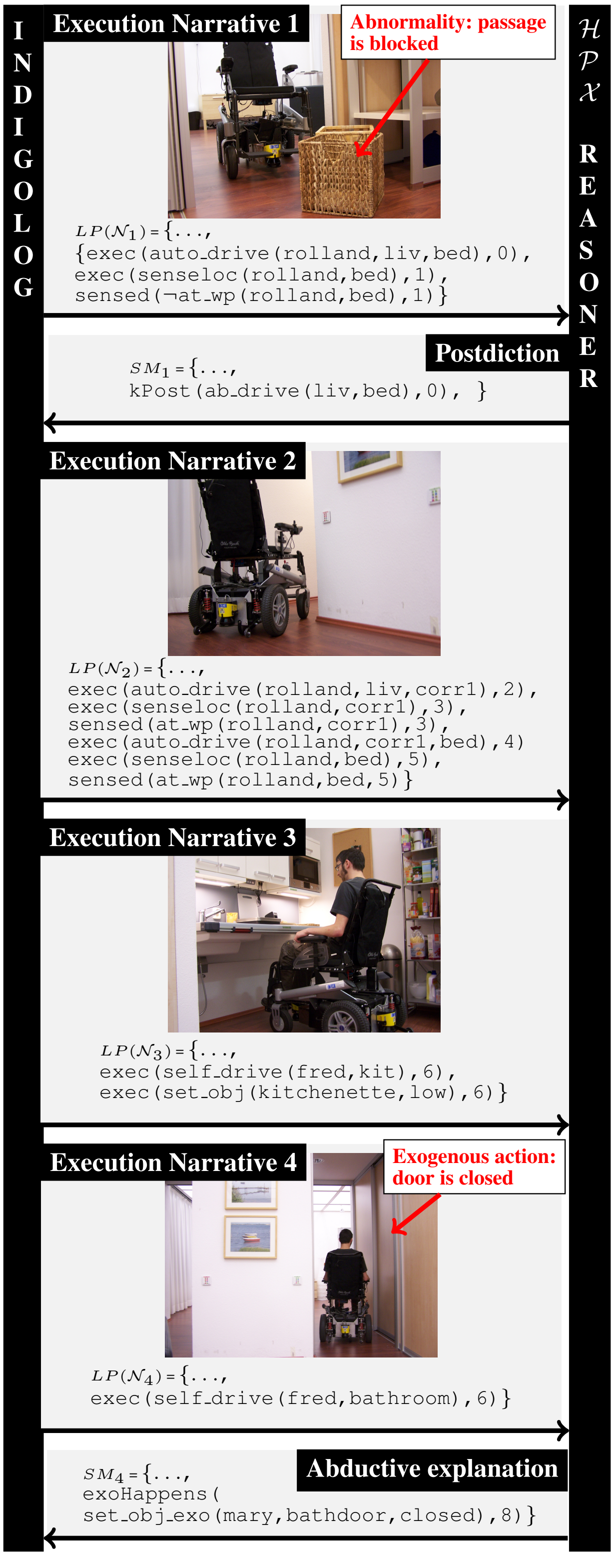


Figure 5. Communication between Indigolog and $\mathcal{HPX}$-reasoner during the scenario

CONCLUSION AND OUTLOOK

The Experimental Commonsense Cognitive Robotics Framework (ExpCog) offers a solid platform for researchers and robotics enthusiasts to easily deploy / experiment / play with tools in commonsense knowledge representation and reasoning and state of the art robotic platforms. ExpCog is aimed at reaching out with mainstream KR methods and tools to researchers at large within other areas of artificial intelligence, especially those concerned with robotics.

We have provided an extensive scenario the demonstrates in particular the interoperation of different reasoners in ExpCog. However, note that ExpCog's modularity makes it very easy to plug in a different robotic platform instead of the BAALL. For example, one could use the same reasoning scheme to postdict abnormal behaviour in an object manipulation task, where the grappler of a robot fails to grab an object due to a broken servo. We have also demonstrated that language elements like causal laws are easily mappable in the case of $\mathcal{HPX}$ and Indigolog, and we aim at enriching our infrastructure with a common domain representation language to further improve the interoperability of different reasoners. Since experiments are a central feature of ExpCog, we explicitly allow and support overlapping reasoning capabilities.

Work is also ongoing for the development and illustration of a broad-based test-suite of problems in cognitive robotics, with a focus on spatial planning and decision-making in both real robotic and simulated environments. The test problems would be used to determine the feasibility of the implemented control approaches and also to perform empirical comparisons amongst them. In addition, they would also be extensively documented from an illustrative viewpoint so as to serve as examples for the utilization of the experimental framework by other users or to be used for educational purposes by teachers and students alike. One test run as a part of a tutorial at the International Spatial Cognition Summer Institute (UCSB, USA) has already been conducted in August 2013, and similar initiatives will be considered regularly. We are presently expanding the user-base of ExpCog, and also plan to release ExpCog publicly for open-source development in due course. Our current ExpCog version is available on request.

The presented status quo of the ExpCog framework is the result of a first consolidation of the independent components that have been pursued in our research on reasoning about *space, actions, and change* [6, 14, 3, 4, 2, 5]. Current research is particularly focussed on developing declarative spatial representation and reasoning methods (e.g., for cognitive vision) that will be embedded within ExpCog, and on further developing the framework keeping in mind challenging new real robot control scenarios / platforms. Information about the current state of ExpCog can be accessed at `www.commonsenserobotics.org`.

ACKNOWLEDGEMENTS

The overall concept for this work originated and benefited from discussions with Maurice Pagnucco (UNSW, Australia). Mehul Bhatt acknowledges the startup funding provided by the Alexander von Humboldt Stiftung (Germany) toward this project. Mehul Bhatt and Jakob Suchan acknowledge funding by the German Research Foundation (DFG). Manfred Eppe acknowledges funding by DFG and the Spanish National Research Council (CSIC). Practical contributions by Sanchit Sood (IIT Bombay, India), Hemant Agrawal (IIT Roorkee, India), Maitreyee Saikia (Tezpur University, India), and Maciej Przybylski (Warsaw University of Technology, Poland) are also acknowledged.

APPENDIX

A BAALL Domain Theory for Indigolog

An Indigolog domain theory consists of two parts: (i) a *domain specification* that contains knowledge about fluents, actions and causal laws and (ii) a *behaviour specification* that governs how an agent acts within the domain. In the following we present a simplified partial specification of the original implementation that we employ for the scenario. In the domain theory, we use variable `Ag` to denote agents, `Wp` for waypoints, `Obj` for objects in BAALL, `Val` for values, `Robo` for autonomous wheelchairs and `Pers` for persons.

Domain Specification

► *Fluents.* To represent the domain, we use fluents for an agent's location (`at_wp/2`) as well as the destination of the robot and the person (`destination/2`). We also use an abnormality fluent `ab_drive/3`, to state that the action of driving to a waypoint has an abnormality, e.g. because the passage is blocked by an obstacle. Additionally we use fluents `has_value/1` to represent the state of objects in the BAALL, such as the setting of the automatic bed or the open-state of a door.

```
prim_fluent(at_wp(Ag, Wp)).
prim_fluent(destination(Ag,Wp)).
prim_fluent(ab_drive(Robo,Wp1,Wp2)).
prim_fluent(has_value(Obj)).
```

► *Actions.* Indigolog supports sensing and non-sensing actions. Both kinds of actions can be primitive, or exogenous. Primitive actions represent action that the Indigolog controller can execute itself, while exogenous actions are executed by external agents. In our case, we use primitive actions to denote e.g. the driving of the wheelchair and exogenous actions to denote that e.g. a person accidentally closes a door in the BAALL.

The main actions are concerned with driving the wheelchair. In `self_drive` mode the person on the wheelchair is controlling it, and in `auto_drive` mode the wheelchair is driving autonomously.

The causal laws describe that if the robot drives, it causes the robot to change its location to the destination of the action if there is no abnormality. In addition, the destination is unset after reaching it. When driven by a person, this person also changes its location.

```
prim_action(auto_drive(Robo, FromWp, ToWp)).
causes_val(auto_drive(Robo,  FromWp, ToWp),
           at_wp(Robo, ToWp), true,
   ab_drive(Robo,FromWp,ToWp)=false)
causes_val(auto_drive(Robo,  FromWp, ToWp),
           at_wp(Robo, FromWp), false,
   ab_drive(Robo,FromWp,ToWp)=false)
causes_val(auto_drive(Robo,  FromWp, ToWp),
           destination(Robo, ToWp), false, true)

prim_action(self_drive(Person, Robo, FromWp, ToWp)).
causes_val(self_drive(Pers, Robo,FromWp, ToWp),
           at_wp(Robo, ToWp), true, true)
causes_val(self_drive(Pers, Robo, FromWp, ToWp),
           at_wp(Robo, FromWp), false, true)
causes_val(self_drive(Pers, Robo, FromWp, ToWp),
           at_wp(Pers, ToWp), true, true)
causes_val(self_drive(Pers, Robo, FromWp, ToWp),
           at_wp(Pers, FromWp), false, true)
causes_val(self_drive(Pers, Robo, FromWp, ToWp),
           destination(Pers, ToWp), false, true)
```

In addition to the wheelchair's driving functionality, we account for actuators in the BAALL environment with the `set_obj` action. Specifically, we involve illumination control, opening and closing doors, as well as adjusting kitchenette, bed and other furnitures. The values of objects can also be set by an exogenous action that involves another agent executing it.

```
prim_action(set_obj(Obj, Val)).
causes_val(set_obj(Obj, Val),
           has_value(Obj), Val, true).

exog_action(set_obj_exo(Ag, Obj, Val)).
causes_val(set_obj_exo(Ag, Obj, Val),
           has_value(Obj), Val, true).
```

In order to verify success of the performed driving actions, the system can sense the wheelchairs location whenever this is required. The gained information is used to trigger the postdictive reasoning that is required for the abnormality detection. Note that postdiction is modeled as an exogenous sensing action; the execution of this action represents that the $\mathcal{HPX}$ reasoner has produced knowledge by postdiction.

```
prim_action(senseloc(Robo, Wp)).
senses(senseloc(Robo,Wp), at_wp(Robo,Wp)).

exog_action(postdict(F)) :- prim_fluent(F).
senses(postdict(F),F) :- prim_fluent(F).
```

► *Initial knowledge.* Indigolog uses the predicate `initially/1` to represent knowledge about initially known fluents. A simplified representation of the illustrated scenario involves knowledge about the destiation and location of wheelchair and person, as well as the state of some objects in BAALL. Note that no knowledge about abnormalities in driving is given, since this will be produced during plan execution by the $\mathcal{HPX}$ reasoner.

```
initially(at_wp(rolland,liv),true).
initially(at_wp(fred,bed),true).
initially(destination(rolland,bed),true).
initially(destination(fred,kit),true).
initially(has_value(kitchenette),low).
initially(has_value(bath_door),open).
initially(has_value(bed_head),low).
```

Behaviour Specification

The behavior specification is realized with Indigolog's efficient *prioritized interrupts*. These can be understood as an ordered list of rules that trigger an action if a condition is known to hold by the Indigolog agent. In the following, we present the implemented interrupts in the order in which they are prioritized.[11]

► *Set bed up if wheelchair approaching.* First, we state an interrupt to move the head part of the bed up if Fred is in the bed and the wheelchair arrives. This simplifies the act of Fred getting onto the wheelchair.

```
interrupt(and(and(at_wp(rolland,bed)=true,
                  at_wp(fred,bed)=true),
                  has_value(bed_head)=low),
   [set_obj(bed_head,high)])
```

► *Drive autonomously.* The wheelchair drives autonomously if it is given a destination and if it is not known that there is an abnormality. In the scenario, Fred is in the bed, where he wants to be picked up. To this end, we implement a corresponding interrupt as follows:

```
interrupt(and(and(at_wp(rolland,liv)=true,
                  destination(rolland,bed)=true),
                  neg(ab_drive(rolland,kit,bed)=true)),
   [auto_drive(rolland,liv,bed),senseloc(rolland,bed)])
```

Note that after the `auto_drive` command, we execute an additional `sense_loc` command to verify whether the wheelchair actually arrived at its destination.

► *Alternative Route.* In the case where there is an abnormality, the wheelchair has to drive an alternative route, in our case via the waypoint `corr1`.

```
interrupt(and(and(at_wp(rolland,liv)=true,
                        destination(rolland,bed)=true),
                        ab_drive(rolland,kit,bed)=true),
 [auto_drive(rolland,liv,corr1),senseloc(rolland,corr1),
  auto_drive(rolland,corr1,bed),senseloc(rolland,bed)])
```

[11] Since the emphasizes of this paper lies in the interoperation of $\mathcal{HPX}$ and Indigolog, and not on the Indigolog implementation itself, we present only a very basic implementation. For example, we do not state general interrupts that fire for all objects of a certain type. However implementing this is straight forward by using Indigolog's ternary `interrupt` directives. We also use a very naive route-replanning mechanism. A more sophisticated method could be implemented with Indigolog's *search* operator.

Note that we execute the `senseloc` command after all driving actions, in order to verify driving success.

▶ *Person-controlled driving.* When the wheelchair arrives at the bed, Fred takes over control and navigates to the kitchen. Since a human is in charge of the actual execution, we do not consider abnormalities for this action.

```
interrupt(and(and(at_wp(fred,bed)=true,
                    destination(fred,kit)=true),
                    at_wp(rolland,bed)=true),
   [self_drive(fred,rolland,bed,kit)])
```

References

[1] Marco Aiello, Ian E. Pratt-Hartmann, and Johan F.A.K. van Benthem, *Handbook of Spatial Logics*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[2] M. Bhatt, H. Guesgen, S. Wölfl, and S. Hazarika, 'Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions', *Spatial Cognition & Computation*, **11**(1), 1–14, (2011).

[3] Mehul Bhatt, 'Reasoning about space, actions and change: A paradigm for applications of spatial reasoning', in *Qualitative Spatial Representation and Reasoning: Trends and Future Directions*. IGI Global, USA, (2012).

[4] Mehul Bhatt, Jae Hee Lee, and Carl Schultz, 'CLP(QS): A Declarative Spatial Reasoning Framework', in *COSIT 2011*, pp. 210–230, (2011).

[5] Mehul Bhatt and Seng Loke, 'Modelling dynamic spatial systems in the situation calculus', *Spatial Cognition and Computation*, **8**(1), 86–130, (2008).

[6] Mehul Bhatt, Jakob Suchan, and Carl Schultz, 'Cognitive Interpretation of Everyday Activities – Toward Perceptual Narrative Based Visuo-Spatial Scene Interpretation', in *Computational Models of Narrative (CMN) 2013., a satellite workshop of CogSci 2013: The 35th meeting of the Cognitive Science Society.*, eds., M. Finlayson, B. Fisseni, Benedikt Loewe, and J. C. Meister, Dagstuhl, Germany, (2013). OpenAccess Series in Informatics (OASIcs).

[7] Michael E. Bratman, *Intention, Plans, and Practical Reason*, Cambridge University Press, March 1987.

[8] A G Cohn, B Bennett, J M Gooday, and N M Gotts, 'Representing and reasoning with qualitative spatial relations about regions', in *Temporal and spatial reasoning*, ed., O Stock. Kluwer, (1997).

[9] A G Cohn and J Renz, 'Qualitative spatial reasoning', in *Handbook of Knowledge Representation*, eds., Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, Elsevier, (2007).

[10] Giuseppe De Giacomo, Yves Lésperance, and Hector J. Levesque, 'ConGolog, A concurrent programming language based on situation calculus', *Artificial Intelligence*, **121**(1–2), 109–169, (2000).

[11] Giuseppe De Giacomo, Yves Lespérance, Hector J. Levesque, and Sebastian Sardina, 'On the semantics of deliberation in IndiGolog – from theory to implementation', in *Proceedings of Principles of Knowledge Representation and Reasoning (KR)*, eds., Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, pp. 603–614, Toulouse, France, (April 2002). Morgan Kaufmann Publishers Inc.

[12] Giuseppe de Giacomo and Hector J. Levesque, 'An incremental interpreter for high-level programs with sensing', *Logical Foundations for Cognitive Agents*, (1999).

[13] Frank Dylla, Lutz Frommberger, and Reinhard Moratz, 'A relative orientation algebra with adjustable granularity', in *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05)*, (2005).

[14] Manfred Eppe and Mehul Bhatt, 'Narrative based Postdictive Reasoning for Cognitive Robotics', in *11th International Symposium on Logical Formalizations of Commonsense Reasoning*, (2013).

[15] Manfred Eppe, Mehul Bhatt, and Frank Dylla, 'Approximate Epistemic Planning with Postdiction as Answer-Set Programming', in *Prodeedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning*, (2013).

[16] R. E. Fikes and N. J. Nilsson, 'Strips: A new approach to the application of theorem proving to problem solving', in *Readings in Planning*, eds., J. Allen, J. Hendler, and A. Tate, 88–97, Kaufmann, San Mateo, CA, (1990).

[17] Andrew U. Frank, 'Qualitative spatial reasoning: Cardinal directions as an example', *International Journal of Geographical Information Science*, **10**(3), 269–290, (1996).

[18] Christian Freksa, 'Using orientation information for qualitative spatial reasoning', in *Proceedings of the Intl. Conf. GIS, From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pp. 162–178, Berlin, (1992). Springer-Verlag.

[19] Martin Gebser, Torsten Grote, Roland Kaminski, Philipp Obermeier, Orkunt Sabuncu, and Torsten Schaub, 'Stream Reasoning with Answer Set Programming: Preliminary Report', in *International Conference on Principles of Knowledge Representation and Reasoning*, (2012).

[20] Michael Gelfond and Vladimir Lifschitz, 'The Stable Model Semantics for Logic Programming', in *Proceedings of the International Conference on Logic Programming (ICLP)*, (1988).

[21] Giuseppe De Giacomo and Hector J. Levesque, 'An incremental interpreter for high-level programs with sensing', in *Logical Foundations for Cognitive Agents*, pp. 86–102. Springer, (1999).

[22] H. Grosskreutz and G. Lakemeyer, 'cc-Golog: Towards More Realistic Logic-Based Robot Controllers', in *NMR-00*, (2000).

[23] Robert Kowalski and Marek Sergot, 'A logic-based calculus of events', *New Gen. Comput.*, **4**(1), 67–95, (1986).

[24] Bernd Krieg-Brückner, Thomas Röfer, Hui Shi, and Bernd Gersdorf, 'Mobility Assistance in the Bremen Ambient Assisted Living Lab', *GeroPsych: The Journal of Gerontopsychology and Geriatric Psychiatry*, **23**(2), 121–130, (June 2010).

[25] Stephane Lallée, Ugo Pattacini, Séverin Lemaignan, Alexander Lenz, Chris Melhuish, Lorenzo Natale, Sergey Skachek, Katharina Hamann, Jasmin Steinwender, Emrah Akin Sisbot, Giorgio Metta, Julien Guitton, Rachid Alami, Matthieu Warnier, Tony Pipe, Felix Warneken, and Peter Ford Dominey, 'Towards a Platform-Independent Cooperative Human Robot Interaction System: III An Architecture for Learning and Executing Actions and Shared Plans', *IEEE Transactions on Autonomous Mental Development*, **4**(3), 239–253, (2012).

[26] Joohyung Lee and Ravi Palla, 'Reformulating the Situation Calculus and the Event Calculus in the General Theory of Stable Models and in Answer Set Programming', *Journal of Artificial Intelligence Research*, **43**, 571–620, (2012).

[27] Hector Levesque and Gerhard Lakemeyer, 'Chapter: Cognitive robotics', in *Handbook of Knowledge Representation*, eds., V. Lifschitz, F. van Harmelen, and F. Porter. Elsevier, (2007).

[28] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl, 'Golog: A logic programming language for dynamic domains.', *J. Log. Program.*, **31**(1-3), 59–83, (1997).

[29] Gerard Ligozat, 'Reasoning about cardinal directions.', *J. Vis. Lang. Comput.*, **9**(1), 23–44, (1998).

[30] Gérard Ligozat, *Qualitative Spatial and Temporal Reasoning*, Wiley-ISTE, 2011.

[31] Jiefei Ma, Rob Miller, Leora Morgenstern, and Theodore Patkos, 'An Epistemic Event Calculus for ASP-based Reasoning About Knowledge of the Past, Present and Future', in *International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, (2013).

[32] Christian Mandel, Kai Huebner, and Tilman Vierhuff, 'Towards an Autonomous Wheelchair: Cognitive Aspects in Service Robotics', in *Towards Autonomous Robotic Systems*, pp. 165–172, (2005).

[33] M Mansouri and F Pecora, 'More Knowledge on the Table: Planning with Space, Time and Resources for Robots', in *ICRA*, (2014).

[34] John McCarthy and Patrick J. Hayes, 'Some philosophical problems from the standpoint of artificial intelligence', in *Machine Intelligence 4*, eds., B. Meltzer and D. Michie, 463–502, Edinburgh University Press, (1969).

[35] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins, 'PDDL – The Planning Domain Definition Language – version 1.2', in *Technical Report CVC TR-98-003, Yale Center for Computational Vision and Control*, (1998).

[36] Reinhard Moratz, Jochen Renz, and Diedrich Wolter, 'Qualitative spatial reasoning about line segments.', in *ECAI*, pp. 234–238, (2000).

[37] Erik T. Mueller, 'Discrete event calculus with branching time', in *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pp. 126–131, (2007).

[38] Martin Peniak, Anthony Morse, and Angelo Cangelosi, 'Aquila 2.0 Software Architecture for Cognitive Robotics', in *Proc. of IEEE International Conference on Development and Learning and on Epigenetic Robotics*, (2013).

[39] David A. Randell, Zhan Cui, and Anthony Cohn, 'A spatial logic based

on regions and connection', in *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, 165–176, Morgan Kaufmann, San Mateo, California, (1992).
[40] Jochen Renz and Bernhard Nebel, 'Qualitative spatial reasoning using constraint calculi', In *Handbook of Spatial Logics* [1], 161–215.
[41] S Rockel, B Neumann, J Zhang, S Konecny, M Mansouri, F Pecora, A M G Saffiotti, A M Tomé, A Pinho, L Seabra Lopez, K S R Dubba, A G Cohn, M Günther, S Stock, J Hertzberg, S von Riegen, and L Hotz, 'An Ontology-based Multi-level Robot Architecture for Learning from Experiences', in *AAAI Spring Symposium*, (2013).
[42] Erik Sandewall, *Features and Fluents (Vol. 1): The Representation of Knowledge about Dynamical Systems*, Oxford University Press, Inc., New York, NY, USA, 1994.
[43] Carl Schultz and Mehul Bhatt, 'Towards a Declarative Spatial Reasoning System', in *ECAI 2012 - 20th European Conference on Artificial Intelligence., Montpellier, France*, (2012).
[44] Moritz Tenorth and Michael Beetz, 'A Unified Representation for Reasoning about Robot Actions , Processes , and their Effects on Objects', *International Conference on Intelligent Robots and Systems*, (2012).
[45] Moritz Tenorth and Michael Beetz, 'KnowRob : A knowledge processing infrastructure for cognition-enabled robots', *International Journal of Robotics Research*, **32**(5), 566–590, (2013).
[46] Michael Thielscher, 'Introduction to the fluent calculus.', *Electron. Trans. Artif. Intell.*, **2**, 179–192, (1998).
[47] Michael Thielscher, 'Flux: A logic programming method for reasoning agents', *Theory Pract. Log. Program.*, **5**(4-5), 533–565, (2005).
[48] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.
[49] Phan Huy Tu, Tran Cao Son, and Chitta Baral, 'Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming', *Theory and Practice of Logic Programming*, **7**(4), 377–450, (2007).
[50] *Handbook of Knowledge Representation (Foundations of Artificial Intelligence)*, eds., Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter, Elsevier Science, December 2007.