SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION

Project funded by the European Community's 7th Framework Programme (FP7-ICT-2011-7)
Grant Agreement ICT–287513

Deliverable D6.3.1

Action generation patterns and search algorithms

Deliverable due date: 31 October 2014	**Actual submission date:** 19 February 2015
Start date of project: 1 November 2011	**Duration:** 48 months
Lead beneficiary for this deliverable: DLR	**Revision:** Final

Nature: P	**Dissemination Level:** CO
R = Report P = Prototype D = Demonstrator O = Other	PU = Public 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)

Table of contents

1 Introduction

In applications involving human-robot interaction, the robot has to accomplish its task while ensuring human safety and reacting to the environment in a compliant manner. Usually, several low-level components such as controllers, safety features, and reactive motion strategies are available in the control architecture. For performing a specific action, e.g. a point to point motion, one has to select suitable components and set their respective parameters. Two or more basic actions may be combined in order to form a complex action pattern, e.g. a pick and place sequence. Such a structure is called an action generation pattern (AGE-P). The creation of an AGE-P aims to encapsulate complexity in order to achieve robustness and to generalise actions while the parameter space is being simplified to an essential and effective usable set.

AGE-Ps may be defined and parameterised directly by experts, who have in-depth knowledge of the task, robot, and control architecture. Alternatively, one may use learning, optimisation, or AI methods that combine basic actions based on certain criteria, e.g. performance or safety. In this deliverable, the approaches by DLR, UNIHB, and IOSB concerning AGE-Ps and search algorithms are described.

2 Safe dynamic motion planning for goal sequences in the presence of dynamic obstacles (IOSB)

A typical task for a mobile robot or a mobile manipulator in an industrial environment is to supply parts to several workstations and collect products from there. Thus, a typical action generation pattern is to move to several goal regions in a pre-specified sequence.

The entire sequence has to be taken into account to achieve (time-) optimal action plans. Generally, neither the optimal position nor the optimal heading for reaching the next goal region may be known in advance. In fact, the optimal way of reaching the next goal region is highly dependent on the subsequent goal. Therefore, an optimal action planning strategy must be able to explicitly take multiple goal regions into account.

Another challenge arises from human-robot interaction in crowded dynamic environments like a collaborative manufacturing scenario. Of course, the robot's plan has to ensure that no collisions with humans can occur. This requires a fast reaction to unexpected motions of humans. A global planning, which is unaware of dynamic obstacles, combined with a subsequent local obstacle avoidance leads to sub-optimal results. Therefore, it is important to consider dynamic obstacles already during the global planning phase. Particularly in narrow spaces, in which robots and humans work side by side, motion planning of mobile robots must also seek to minimise the collision risk as much as possible. This applies to the close vicinity of the robot (risk of collision due to dynamic obstacles) as well as to the farther regions (parts of the environment with increased collision risk, e.g., near static obstacles or common routes of human co-workers).

Fraunhofer IOSB has developed a novel concept which solves the action planning problem in the context of narrow spaces that potentially contain many dynamic objects. In order to keep this complex task computationally tractable, we employ a state-time lattice with variable dimensionality to exploit the fact that the required accuracy of the solution in the near range of the robot is higher than it is in farther regions. Our planning algorithm provides a solution of hybrid nature. In the immediate future, it explicitly contains a time component, thus providing a trajectory. With progressing time, we relax the requirements on the solution, so that it represents merely a kinematically feasible path. Furthermore, the planning is done collectively for the next two goal regions to allow the robot to arrive at the first goal region in an optimal manner with respect to its subsequent travel. We model the static and dynamic risk of the environment in a consistent way. This allows the joint planning for multiple goals while incorporating collision risk due to dynamic and static obstacles.

The lattice is composed of action primitives representing motions which are kinematically and dynamically feasible for the robot. We employ a multi-resolution approach which achieves a high computational performance due to a coarse planning in free spaces, while being able to refine the resolution if necessary in narrow spaces and in the vicinity of obstacles. The search for a safe action sequence within the lattice is performed by the Anytime Repairing A* algorithm (ARA*) [1]. This algorithm quickly finds a suboptimal solution and then iteratively refines it towards the optimal one if sufficient computation time is available. In this way, safety is guaranteed by a fast reaction to unexpected changes in the environment, while the optimality of the computed action plans is achieved in the majority of the cases.

The action planning algorithm has been successfully applied to a mobile robot in real-time. Figure 1 shows an example of a planned path/trajectory. The robot (green) faces an oncoming obstacle (red) and plans an avoidance manoeuvre according to the predicted motion of the obstacle. The red ellipses depict the obstacle's uncertainty which increases with time.

Figure 2 illustrates the application of this action generation pattern to a sequence of two goals.

For more details, please refer to the following publications:

- J. Petereit, T. Emter, C. W. Frey, "Safe Mobile Robot Motion Planning for Waypoint Sequences in a Dynamic Environment", IEEE International Conference on Industrial Technology (ICIT), 2013.
- J. Petereit, T. Emter, C. W. Frey, "Mobile Robot Motion Planning in Multi-Resolution Lattices with Hybrid Dimensionality", IFAC Intelligent Autonomous Vehicles Symposium, 2013.
- J. Petereit, T. Emter, C. W. Frey, "Combined trajectory generation and path planning for mobile robots using lattices with hybrid dimensionality", 2nd International Conference on Robot Intelligence Technology and Applications (RITA), 2013

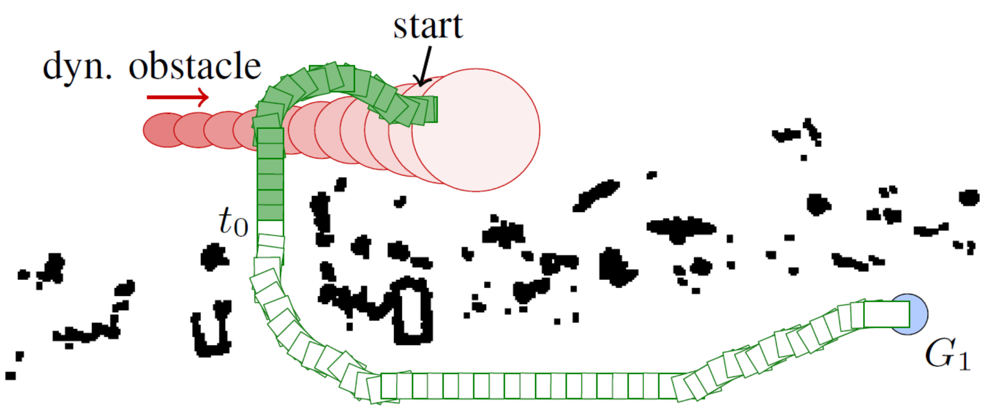


Figure 1: Hybrid path and trajectory planning for a mobile robot in the presence of dynamic obstacles.

Figure 2: Mobile robot moving to two subsequent goals: (a) separate planning towards each goal, (b) combined planning for the action sequence composed of two goal regions. As expected, the combined planning is superior to the separate planning as it can avoid the unnecessary turning manoeuvre.

3 Sensor-based Grasping of Objects (IOSB)

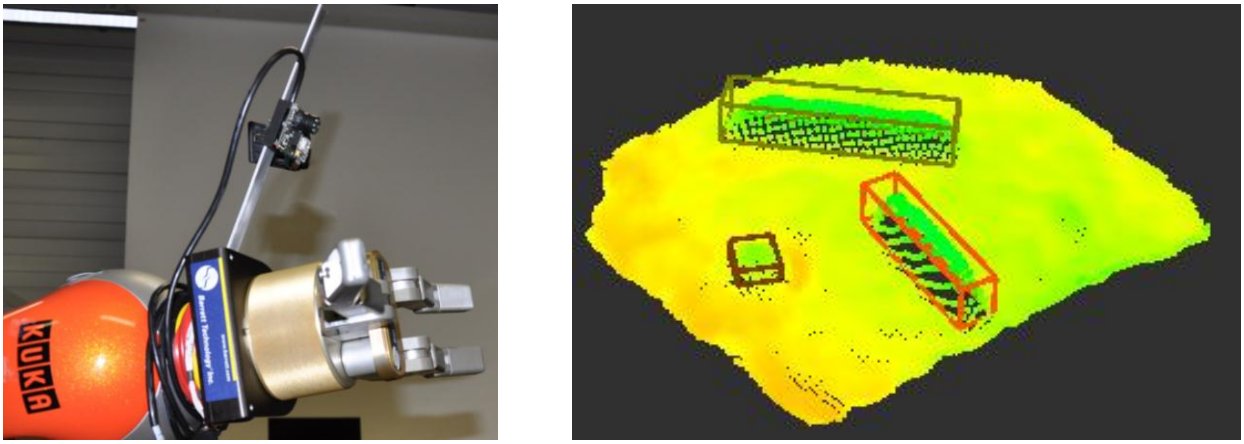
Another action generation pattern implemented by IOSB is concerned with grasping objects by a robotic arm. Such grasping tasks occur frequently in the SAPHARI use cases, where the mobile manipulator has to supply parts to workstations, for instance. The task under consideration is to grasp box-shaped objects located on a planar surface like a table or a shelf. The dimensions and the orientation of the object are unknown. Furthermore, its position is not known exactly. Thus, the actions have to be adapted based on sensor information.

To demonstrate the grasping actions, the mobile platform omniRob has been equipped with a LWR manipulator, a Barrett Hand and a PMD CamBoard nano depth camera (see Figure 3 a)). The depth camera is mounted close to the tool centre point of the manipulator so that it can observe the object to be grasped. Based on the acquired 3D point cloud, objects located on top of the planar surface are detected. A bounding box model of the objects is estimated (see Figure 3 b)). Considering the estimated model and the limitations imposed by the geometry of the hand, suitable grasping points are selected.

The action generation pattern is composed of a sequence of states (Figure 4): First, the robot arm moves to a position above the table so that it can observe the region of interest in which the object is supposed to be located. Once the object is detected, the hand is positioned exactly above the object and oriented according to the computed grasping points (Figure 5). Then, the robot can move its hand down towards the object without the risk of touching the object accidentally with its fingers. Finally, the fingers are closed to grasp the object. The robot may then be instructed to place the object on another table. Therefore, it moves exactly above the desired placing position and then moves down to the table. The height of the table is determined from the depth camera data. The hand releases the object and finally, the robot moves upwards in order to avoid hitting the object with its fingers.

During the whole action sequence, collision-free robot paths are planned by a RRT motion planner using a 3D scene model of the environment. Furthermore, workspace monitoring is active in order to prevent collisions with dynamic obstacles. For instance, the robot is slowed down or stopped if a human crosses its path.

Many action parameters have to be estimated online during execution, e.g., dimensions and orientation of the object, height of the table, grasping points. These parameters may even change when the execution of the atomic action has already been started, as the object localisation improves once the camera gets closer to the object, for example. If some atomic action fails, the robot needs to retry with different parameters or to fall back to an earlier action (these state transitions are omitted in Figure 4 for clarity). For example, the camera is moved to a different position if no object is visible.



a) b)

Figure 3: a) Setup of robot hand and PMD camera, b) acquired 3D point cloud and estimated bounding box models of the objects located on the planar surface.

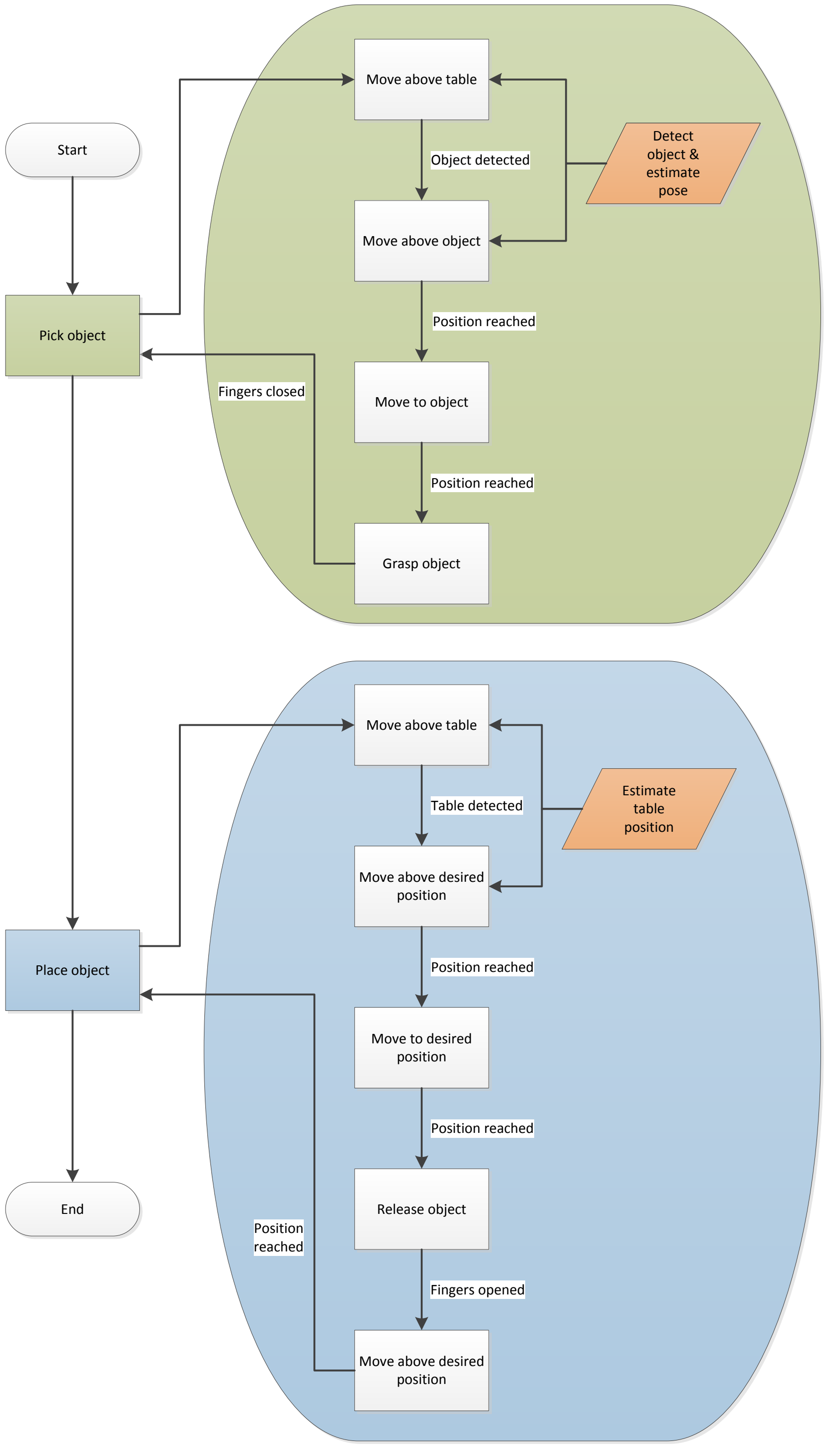


Figure 4: State sequence of the action generation pattern for grasping and placing objects.

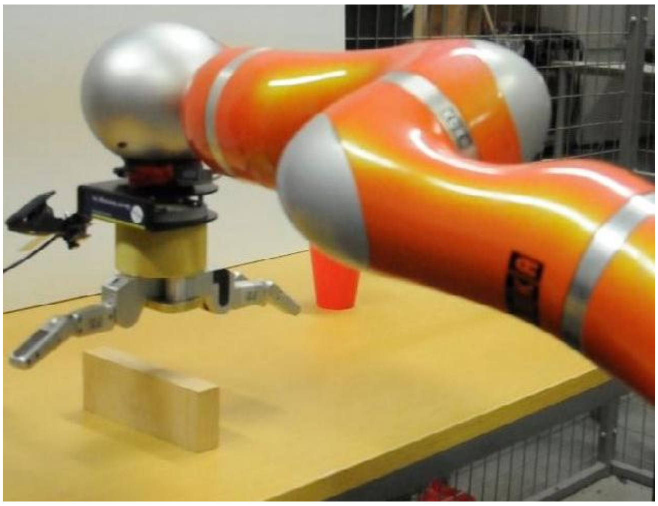


Figure 5: Positioning the robot hand exactly above the object, with fingers oriented according to the chosen grasping points.

4 Safe action planning for a manipulator in the presence of humans (DLR)

4.1 AGE-Ps/Action graphical representations and libraries

In order to program tasks involving manipulation and interaction for complex robots such as the LWRIII, it is important to be able to easily integrate new planning and perception components. Furthermore, robot programming needs to be intuitive but yet powerful and flexible. The simple programming of reactive action generation patterns and their encapsulation is a highly desirable feature for reuse of already designed control programs. We have developed a robot programming software that allows for designing control programs and distributed computation on various levels of abstractions and, if desired, with various underlying paradigms. Originating from early state based approaches with statically compiled hybrid state machines (Figure 6 (left)), we designed the dynamically programmable framework "Bubbles", see Figure 6 (right).

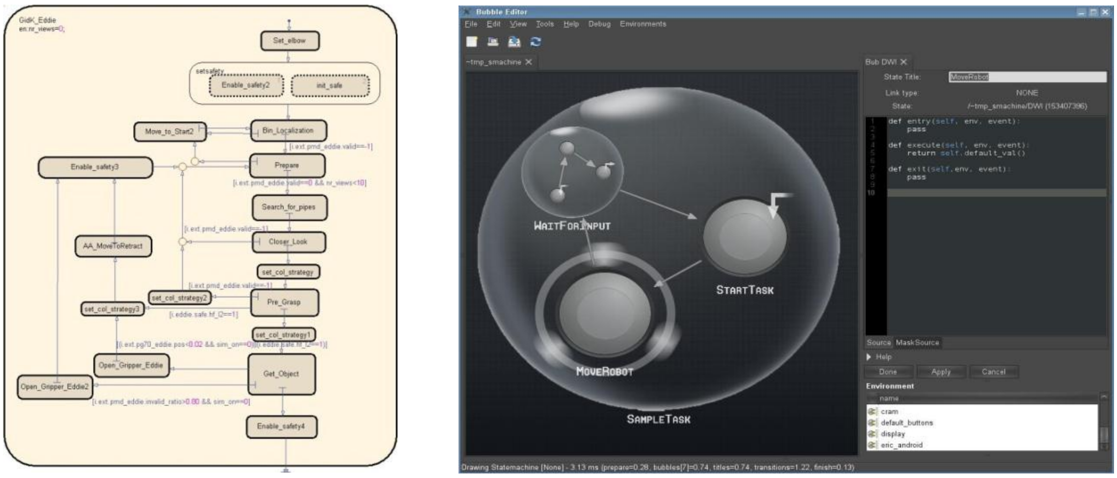


Figure 6: Early Bubbles based on static hybrid state machine design (left), current prototype of bubbles (right).

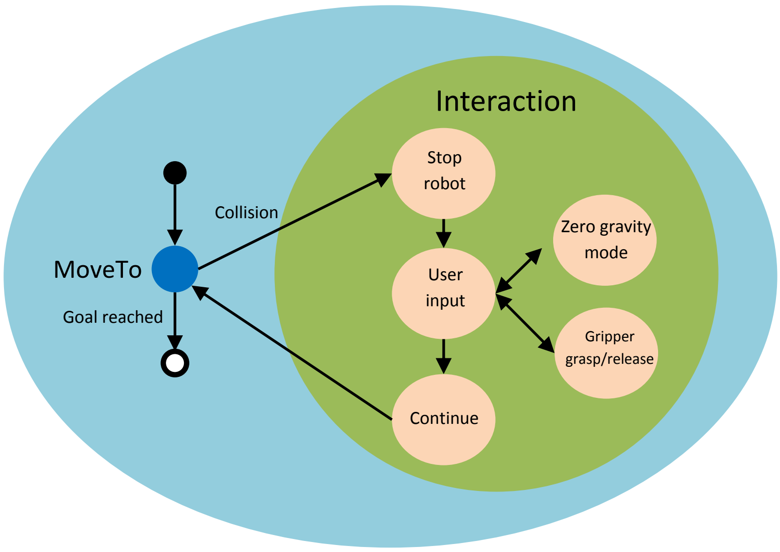
It allows access to the full functionality of all attached system components via Python scripting. It provides a full library mechanism for encapsulating invariant action patterns (or skills) and provides them to the programmer for flexible reuse and task dependent parameterisation. The underlying programming paradigm is an automaton-based approach, which allows an intuitive understanding of the program structure. Bubbles supports parallelism, seamless hierarchy, and flexible integration and communication of external components such as sensors, planners, or observers. Another important feature of the system is the ability to change programs at runtime, either based on a planner or actively by the user. This allows an online development of tasks while the robot executes the current program. However, apart from serving as a programming tool, Bubbles acts also as a graphically programmable planner that allows for optimal interaction with the real-time control framework BEASTY [2], which was already made available to UNIHB. In particular, it allows the combination of expert based program snippets and planners and collaboration with true AI-planning tools.

We use basic actions to command the BEASTY framework, which are the very principle commands that are sent to the RCU in order to create and execute AGE-Ps. In this concept, basic actions like an "atomic movement" or "stop" are enclosed and commanded with an additional behaviour parameter set to the RCU. The behaviour defines which controller, interpolator and virtual impedance parameterisation is used. The strict separation of robot commands and the corresponding robot behaviour parameter sets offer a user friendly programming of the task and were also used to formalise human-robot interaction [3]. Nevertheless, the programmer still needs to have a reasonable knowledge of the control methods to program basic actions that do not lead to dangerous situations or error states.

By combining two or more basic actions one can create an AGE-P, which constitutes a formal structure to fulfil a complex task and reduces the need of expert knowledge by a possibly small set of symbolic controller parameters covering all task- and safety-related action aspects. To generate an AGE-P our setup provides amongst others following components:

- Controller settings:
    - Interpolator settings
    - Zero gravity control
    - Impedance parameters
- Trajectory generation:
    - Collision avoidance
    - Tactile exploration
    - Time scaling
- Environment observers:
    - Collision detection
    - Human observation
    - Workspace observation
- Safety features:
    - Collision reaction
    - Collision classification
    - Safe velocity control

This combination of library automata that encapsulate certain complex action pattern, with task or interaction planners that may then reason about these more complex AGE-Ps, is a powerful approach to combine expert knowledge and AI-planning techniques. In the following, we describe an AGE-P existing in Bubbles in more detail, namely the "MoveAttentive" pattern, see Figure 7.



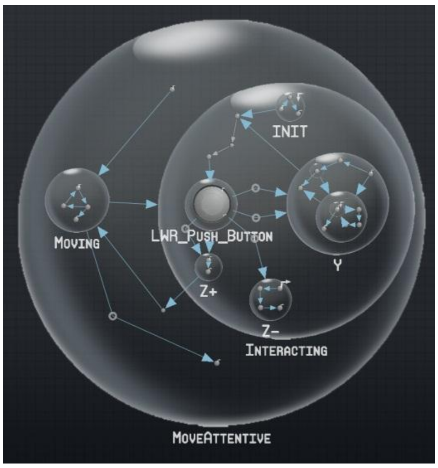


Figure 7: "MoveAttentive": A library planning module that allows to program "move to"-commands such that they are inherently equipped with interaction patterns, fault behaviours, and re-entering schemes. In the left figure, the abstract structure is depicted while the Bubbles implementation is shown in the right figure.

The so called "MoveAttentive" AGE-P extends the simple "MoveTo" Atomic Action and allows various types of physical human-robot interaction and task recovery strategies. If the robot does not collide with the environment while moving, the "MoveAttentive" AGE-P will simply behave like a MoveTo. If, however, there is physical contact, the robot will stop and enter an interaction mode. Here, the person can decide on the next action, which is either

- Continue the task,
- Open/close the gripper, or
- Activate zero gravity control.

The transitions between all states are well defined, the according Bubbles control structure is depicted in Figure 7 (right).

To demonstrate the "MoveAttentive" AGE-P, we select a simple pick and place scenario. The setup consists of a DLR LWRIII, which is equipped with a pneumatic gripper; see Figure 8 (left). The goal of the task is to pick chocolate bars from the table and place them onto a 6x6 field.

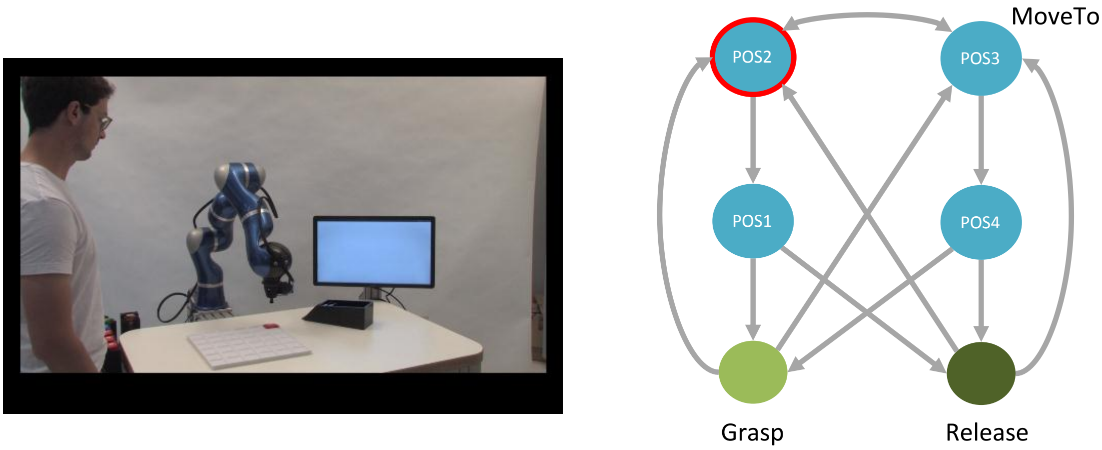
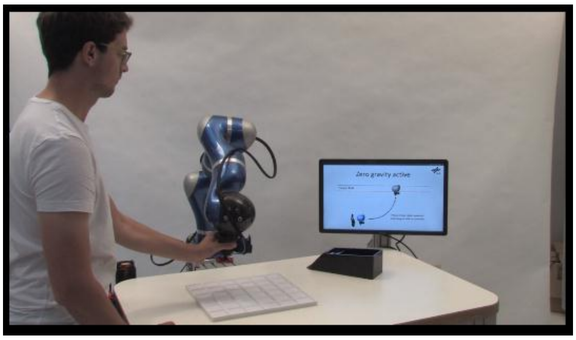


Figure 8: Test setup (left), robot program using Atomic Actions (right).

First, the task is programmed using the Atomic Actions "MoveTo", "Grasp" and "Release", see Figure 8 (right). As mentioned above, the robot can only accomplish the task if the conditions are perfect. It is not possible to react to changes in the environment reasonably. In the second part of the experiment, we simply replace the "MoveTo" Atomic Actions with "MoveAttentive" AGE-Ps, see Figure 9 (right). Please note that the overall high-level structure remains the same. This ensures an intuitive, easy to use programming framework.



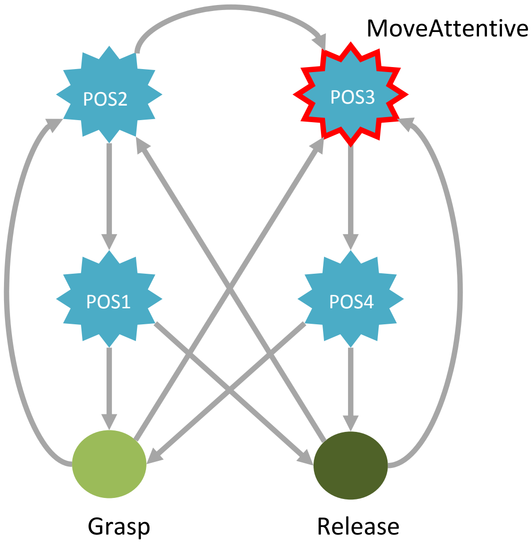


Figure 9: Human-robot interaction (left), robot program using AGE-Ps (right).

After touching the robot, the interaction mode depicted in Figure 9 is being activated. Multimodal audio and visual feedback (screen) is provided to help the user decide on the next robot action. In Figure 9 (left) the human is moving the robot, which is being controlled in a compliant zero gravity mode. After leaving the interaction mode, the robot continues its task.

4.2 Multi-agent reinforcement learning for task learning and adaption

In order to be able to learn and modify robot control programs, DLR investigated the domains of dynamic action and behaviour learning, modification, and selection for complex robotic systems that are sought to act in dynamic and partially unknown environments. In this regard, an algorithmic framework for learning high-dimensional, interactive robot actions was developed. The framework is based on an extended version of optimal adaptive learning for extensive support of dynamic, however, still human-friendly action generation. Figure 10 depicts a high-level schematic view of the framework.

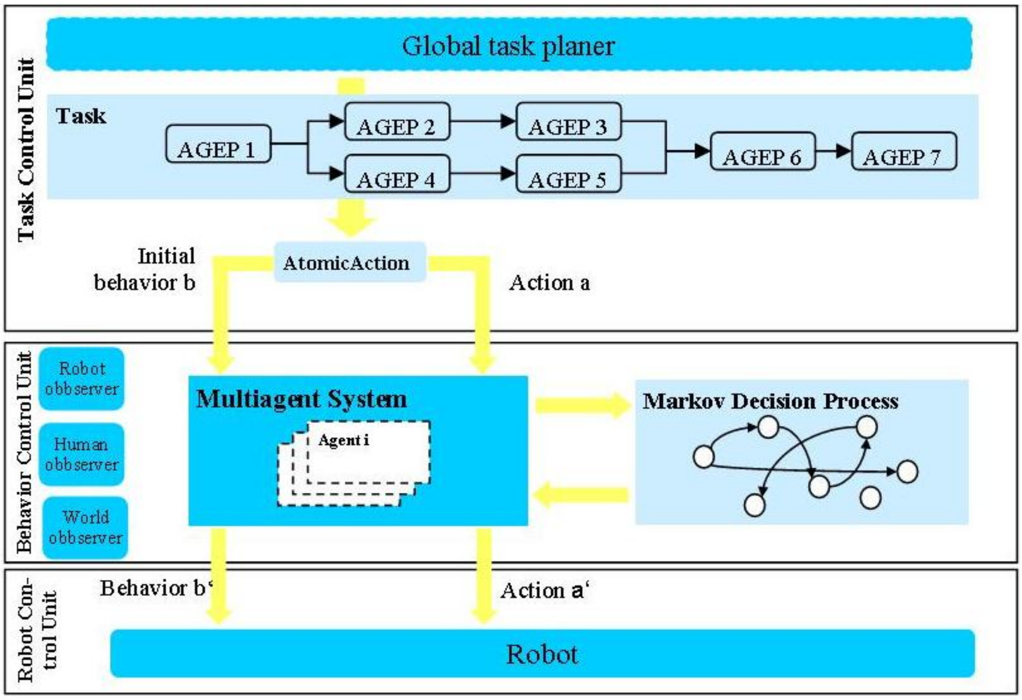


Figure 10: Algorithmic reactive control, learning, and adaptation framework.

Every agent represents a specific part of the robot behaviour or action, influences the behaviour of the multi-agent system and incorporates observed events and information from the robot, humans and the surrounding environment.

The schemes utilise a concept for modelling interaction based on an interaction world and safety related metrics [4]. In addition, we design an online behaviour selection and adaptation algorithm that enables the robot to locally adapt its behaviour such that human safety can be ensured in case of undesired and potentially dangerous events. The developed framework intends to bridge the gap between non-real-time task planners and hard real-time robot control algorithms for complex robotic systems.

This learning and adaptation framework was demonstrated to the consortium by showing the task "Put Bottle Away" during the Y1 review meeting, see Figure 11. The task was part of the test use case "Clean Up Desk", which was situated in a medical context. A bottle is to be placed into storage by an AGE-P consisting of the actions "Move" and "Open Gripper". This AGE-P is equipped with four additional actions, namely "Stop", "Move to Safe Position", "Handover" and "DropBottle", which allow an adaptation of the nominal robot behaviour. The respective task state machine is depicted in Figure 12.

Figure 11: Hand-over scenario.

By executing the task repeatedly and integrating human feedback, the Markov agents were trained within a few cycles. The agents were then able to react to changes in the environment immediately, e.g. a human entering the robot's workspace, and adapt the task sequence accordingly.

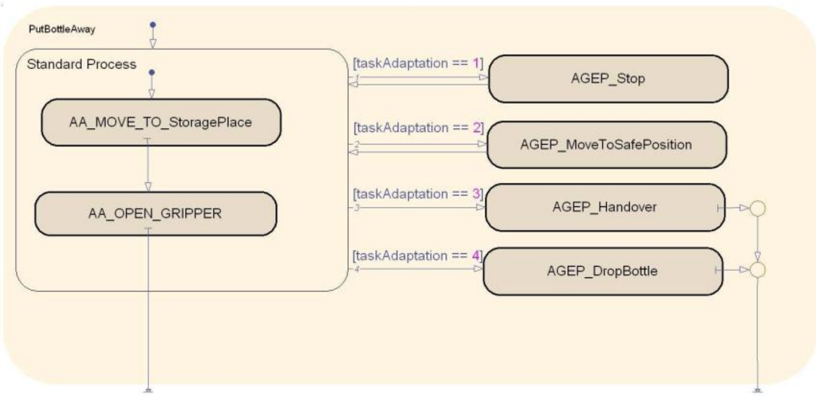


Figure 12: Example of the AGE-P "Put bottle away".

5 Using safety-aware robotic agents to solve abstract and underspecified representation of action (UNIHB)

The research efforts of University Bremen (UNIHB) towards action generation patterns and the corresponding action search algorithms (AGE-Ps) centred on extending its existing software frameworks CRAM and KnowRob. More specifically, first-class representations of motions, humans, and safety-related events were added to both frameworks. Using these explicit representations, UNIHB investigated reasoning methods which select and parameterise the safety features of the safety-aware motion controller BEASTY, which was provided by DLR. To introduce and describe their conceptual integration apparatus of safety-aware robotic agents, DLR and UNIHB wrote a joint paper which is currently under review [5].

In its research, UNIHB investigates knowledge- and cognition-enabled plan-based control systems for autonomous robots. With regard to software architecture, the framework CRAM acts as the task executive module which follows the paradigm of cognition-enabled plan-based control. The software toolbox KnowRob, on the other hand, acts as the knowledge base of the system providing on-demand knowledge processing and information interpretation to CRAM.

At the beginning of the SAPHARI project, CRAM supported vague first-class representations so called designators for objects, locations, actions, and goals. During task execution, the system employed cognitive algorithms to ground its abstract designators in the belief state of the robot and search for the correct action selection and parameterisation in a given execution context. With SAPHARI support, UNIHB published a paper detailing the abstraction and failure handling capabilities of CRAM in a robotic pick and place scenario [9]. Furthermore, UNIHB used SAPHARI funding to extend CRAM with a temporal projection reasoning system which searches through the space of possible action parameterisations by executing them in a physics-based belief state module [8].

As the SAPHARI project works towards safe physical interactions of robots with humans, UNIHB decided to extend its high-level robot control systems with first-class representations of robot motions. The overall conceptual approach and rational is described and justified in detail in the joint paper with DLR [5]. In short, we argue that high-level task executives which shall decide on a human-safe course of actions for robots need to have the representational possibility to reason about its motions because it is the movements of the robot which eventually may cause harm to a human co-worker. As a result, UNIHB extended CRAM and KnowRob with representations, execution strategies, and reasoning mechanisms for robotic motions for sophisticated activities such as pouring of liquids and pancake flipping [7,6].

In its ongoing but still unpublished work, UNIHB has extended the semantic robot description language (SRDL) to also cover human agents. Previously, CRAM and KnowRob employed SRDL to reason about the capabilities of the robotic agents. Now, both modules have access to a human SRDL to describe and react to the capabilities of human co-worker in a given situation. Additionally, UNIHB worked on modelling the BEASTY controller specification as detailed by partner DLR in [2] using the prolog logic programming language of KnowRob. In short, a BEASTY Cartesian motion goal consists of the Cartesian goal pose, velocity and acceleration thresholds for each goal dimension, Cartesian Impedance stiffnesses and damping, safety reactions to external collisions, virtual repelling spheres, and virtual repelling walls.

Regarding AGE-Ps, a CRAM plan which achieves a given action goal constitutes an abstract and underspecified representation of this action. This plan can be generated from and searched for in the KnowRob ontology of robotic actions. Within such a plan vague location descriptions designate motion goals which the executive resolves using the sampling-based search mechanisms presented in [8]. Additionally, KnowRob has ontologies for motions [6] and safety situations, e.g. "a robot motion to make contact/release contact", "a movement with/without human presence", and "a contact initiated by the human/robot", with associated thresholds and stiffness values. At runtime, the CRAM task executive searches for the appropriate motion [6] and safety classes in the KnowRob ontologies, effectively resolving [9] its abstract movement description using the currently perceived state of the human co-worker and task execution context. The system parametrises the BEASTY controller with the threshold and Impedance values associated with the respective safety motion classes. Finally, the abstract human designators are resolved using the perception system, and the collision shapes of the perceived body parts complete the fully-specified controller parameterisation. UNIHB has evaluated the prototypes of its system in the hospital scenario [5] and will continue using this demonstrator for evaluation purposes in the SAPHARI project.

6 Summary

Within the SAPHARI project, several approaches were taken to design and parameterise AGE-Ps. At DLR, the IDE Bubbles is used to combine basic actions to form AGE-Ps. Firstly, AGE-Ps can be manually designed by experts, which however requires deep knowledge about the robot and the task. In order to take the environment and human feedback into account, a learning framework was developed that selects the most appropriate action sequence based on safety metrics. Having once trained the learning framework, no additional computation time is required to select and parameterise action patterns during task execution. This was demonstrated by DLR in a medical setting.

For motion planning, IOSB has employed a multi-resolution approach in order to find a kinematically and dynamically feasible sequence of goal states, while taking dynamic obstacles in the environment into account. By using search algorithms such as A*, ARA*, and RRT, large search spaces can be covered. Accurate motions are computed in the vicinity of the robot, coarse motion plans are found for farther regions. Furthermore, an AGE-P was designed for grasping an object. The AGE-P considers dynamic changes in the environment and performs retry strategies in case a grasp has failed.

The plan-based controller CRAM developed by UNIHB can generate and search action goals from the knowledge processing system KnowRob. Sampling-based search mechanisms are then used to resolve the abstract plans. Finally, the system parametrises the low-level BEASTY controller with the threshold and impedance values associated with the respective safety motion classes.

The practical feasibility of the proposed action generation patterns and search strategies has been shown in several experiments and demonstrations. Furthermore, most of the methods described in this deliverable will be implemented in one of the WP8 use cases. Distinct differences can be identified among the approaches. On the one hand, this allows comparing the advantages and disadvantages of the methods. On the other hand, the approaches may be combined in the future to improve the system performance in terms of safety, task fulfilment, and usability.

References

[1] M. Likhachev, G. Gordon, and S. Thrun: ARA*: Anytime A* search with provable bounds on sub-optimality. *Proceedings of the Conference on Neural Information Processing Systems*, 2003.

[2] S. Parusel, S. Haddadin, and A. Albu-Schäffer, Modular state-based behavior control for safe human-robot interaction: A lightweight control architecture for a lightweight robot, *International Conference on Robotics and Automation*, 2011.

[3] H. Widmoser, Interaction Planning for collaborative Human-Robot Assembly Tasks, Master's Thesis, *DLR-Internal Report DLR-IB 572-2012/3*, 2012.

[4] M. Englert, Optimal search strategies for selecting human-friendly robot behaviors in near-realtime, *DLR-Internal Report*, 2011.

[5] M. Beetz, G.Bartels, A. Albu-Schäffer, F. Bálint-Benczédi, R. Belder, S. Haddadin, A. Maldonado, N. Mansfeld, T. Wiedemeyer, R. Weitschat, and J. Worch, Robotic Agents Capable of Natural and Safe Physical Interaction with Human Co-workers, *International Conference on Autonomous Agents & Multiagent Systems*, 2015, submitted for review.

[6] M. Tenorth, G.Bartels, and M. Beetz, Knowledge-based Specification of Robot Motions, *European Conference on Artificial Intelligence*, 2014.

[7] G. Bartels, I. Kresse, and M. Beetz, Constraint-based Movement Representation grounded in Geometric Features, *IEEE-RAS International Conference on Humanoid Robots*, 2013.

[8] L. Mösenlechner and M. Beetz, Fast Temporal Projection Using Accurate Physics-Based Geometric Reasoning, *IEEE International Conference on Robotics and Automation*, 2013.

[9] J. Winkler, G. Bartels, L. Mösenlechner, and M. Beetz, Knowledge Enabled High-Level Task Abstraction and Execution, *Advances in Cognitive Systems*, 2012.