

Robotic Agents Capable of Natural and Safe Physical Interaction with Human Co-workers

Michael Beetz^{*}, Georg Bartels^{*}, Alin Albu-Schäffer⁺, Ferenc Bálint-Benczédi^{*},
Rico Belder⁺, Daniel Beßler^{*}, Sami Haddadin[#], Alexis Maldonado^{*},
Nico Mansfeld⁺, Thiemo Wiedemeyer^{*}, Roman Weitschat⁺, Jan-Hendrik Worch^{*}

Abstract—Many future application scenarios of robotics envision robotic agents to be in close physical interaction with humans: On the factory floor, robotic agents shall support their human co-workers with the dull and health threatening parts of their jobs. In their homes, robotic agents shall enable people to stay independent, even if they have disabilities that require physical help in their daily life – a pressing need for our aging societies. A key requirement for such robotic agents is that they are safety-aware, that is, that they know when actions may hurt or threaten humans and actively refrain from performing them.

Safe robot control systems are a current research focus in control theory. The control system designs, however, are a bit paranoid: programmers build “software fences” around people, effectively preventing physical interactions. To physically interact in a competent manner robotic agents have to reason about the task context, the human, and her intentions. In this paper, we propose to extend cognition-enabled robot control by introducing humans, physical interaction events, and safe movements as first class objects into the plan language.

We show the power of the safety-aware control approach in a real-world scenario with a leading-edge autonomous manipulation platform. Finally, we share our experimental recordings through an online knowledge processing system, and invite the reader to explore the data with queries based on the concepts discussed in this paper.

I. INTRODUCTION

Imagine a robotic agent working as an assistant in a hospital, helping with tasks such as tidying up or arranging surgical instruments on a tray prior to operations. While the robot holds a scalpel to put it onto the tray, a human co-worker suddenly steps into its reach – Figure 1 depicts a similar scene. As robot system designers, we would like the robot control program to identify this as a potentially dangerous situation, and react by a) pointing the sharp blade of the scalpel away from the human, and b) stopping or drastically reducing speed and stiffness of its motion.

After halting, the arm of the robot blocks the view of the human onto the surgical instruments on the table. As she is looking for an instrument on the table, she steps even closer and tries to push the arm of the robot out of view. The robot control program should react to this strong but purposeful

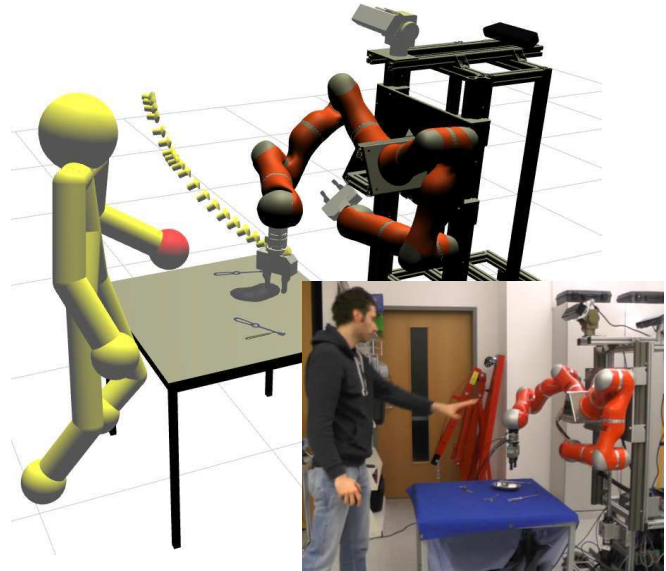


Fig. 1. Safety aware robotic agents know when actions have the potential to hurt or threaten humans and they actively refrain from performing them. To build such an agent, we extended the conceptual apparatus of cognition-enabled robot control with first-class representations of humans, motions, and safety events. Here we visualize from the logged belief state of the robot, which postures it believed itself and the human co-worker had, where it saw objects, and which motion it performed when a human entered its workspace.

collision by switching to gravity compensation mode to yield to the intentions of the human.

Consider a further episode: The robotic assistant tries to hand over a retractor to its human co-worker. To make the tool easier to grasp, the robot control program should orient the instrument handle-first towards the human. During the hand-over the motion control system of the arm reports a slight collision. The robot control program should expect small collisions at the end of hand-overs when human and robotic hands are closer, and ignore them or expect them as indication of a successful hand-over.

We call a robotic agent capable of exhibiting such competent behavior *safety-aware*. A safety-aware agent knows when an action has the potential to hurt or threaten a human co-worker, and it actively refrains from performing this action. We believe that, first and foremost, any robot which shall work in close proximity or direct contact with humans has to be safety-aware. Additionally, it should also *naturally*

^{*}The author is with the Institute for Artificial Intelligence, Universität Bremen, Am Fallturm 1, 28359 Bremen, Germany.

⁺The author is with the German Aerospace Center, Institute of Robotics and Mechatronics, Münchner Straße 20, 82234 Wessling, Germany.

[#]The author is with the Institute of Automatic Control, Leibniz Universität Hannover, Appelstraße. 11, 30167 Hannover, Germany

employ physical interaction when appropriate or necessary, i.e. not paranoidly avoid physical interactions per design.

Current robotic manipulation systems already contain the blocks necessary to build safety-aware robotic agents. State of the art motion controllers can move fast, slow, with high or low impedance, detect collisions, and change into other control modes within a single control cycle [1]. Besides this, there are numerous vision algorithms for detecting and tracking human bodies in color and depth images [2], [3], [4]. Robot control programs lack, however, basic knowledge about tasks, humans and motions to act competently and safely in open human environments.

In this paper, we present an extension to the framework of *cognition-enabled robot control* [5] in which we represent the human co-worker, the robot’s motions, and the safety-related physical interaction events explicitly as first-class objects in the plan language. These new programming concepts allow us to build robotic agents which can reason about their motions and observed human co-workers in terms of related safety events. At the end of the paper, we show the efficacy of our safety-aware approach to robot control in a real-world experiment with a leading-edge autonomous mobile manipulation platform performing a pick and place tasks with surgical instruments while at the same maintaining the safety of its actions w.r.t. to an interfering human co-worker.

II. CONCEPTUAL APPROACH

Let us now introduce the foundations of safety-aware control, the cognition-enabled control framework, and sketch how we will extend the framework to facilitate safety-aware control.

A. Conceptual Foundations

The key issue in safety-aware physical human-robot interaction is that plans have to specify how the robot should perform its motions in a safety-aware manner. This requires the robot to (1) perceive safety-relevant entities and events and parameterize its motions safely and (2) reason about safety aspects and their consequences on the robot’s physical actions. While the former aspect is typically (partly) addressed in the research field of control theory, the latter is a core problem in artificial intelligence (AI). We will briefly introduce the concepts of both fields needed to realize safety-aware robotic agents.

1) *Control Systems Engineering Concepts*: Control theory provides the conceptual apparatus to mathematically model and analyze the properties of dynamical systems. Using these tools, one can build feedback controllers which specify the signals to the motors of the robot to achieve a desired physical movement.

A typical motion controller for a robot continually monitors its dynamical state, compares the sensed state with the desired motion trajectory, and issues commands to minimize the error between the sensed and desired motion trajectory.

The dynamical state of the system is modeled in terms of state variables typically including the joint positions, joint torques, and their derivatives. The control system might also

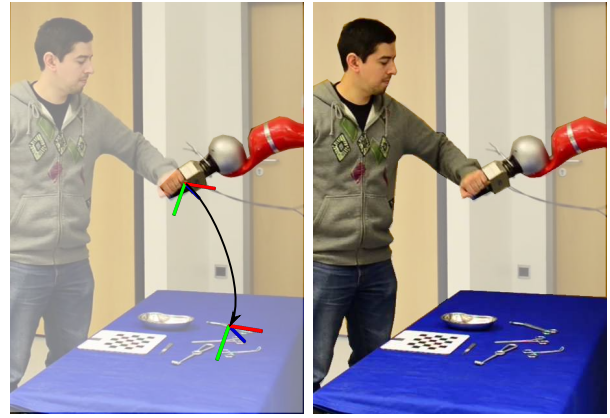


Fig. 2. Left: Control approaches specify motions using Cartesian end-effector attractor frames and repelling spherical and planar force fields. Right: Symbolic approaches logically represent objects, humans, and robots w.r.t. background knowledge. Actions axiomatically relate declarative pre-conditions, e.g. the gripper is empty and the clamp is on the table, to their effects, e.g. the clamp is held in the gripper. In this paper, we address how to bridge this representational gap in the context of safety-aware robotic agents.

express its state w.r.t. to reference frames attached to parts of the robot (like the end-effector) or other relevant objects. The range of frameworks for expressing desired external motion trajectories ranges from basic forward and inverse kinematics to advanced formalisms like iTASC [6].

Using the deep mathematical models of the kinematics and dynamics of the robotic manipulator programmers can often ensure the stability and sometimes even the optimality of their controllers.

While the conceptual apparatus of control engineering is very expressive in terms of formulating movements it does not allow us to talk about the context in which are motions to be performed, e.g. the presence of a human, and the effects of movements on object and agent states.

2) *AI Models of Action Control*: Symbolic approaches to action control, on the other hand, enable the programmers to specify tasks in terms of objects that are to be manipulated and used, the desired effects and unwanted side effects. They allow to specify actions such as “give the mug to the person”, which can be translated into “place the mug into the open hand of the person from above, and release on contact”.

As a consequence, the robot can reason about what it is doing and why. In short, symbolic approaches typically employ abstract models of actions that represent actions in terms of their preconditions and effects. However, these models abstract away how the agent shall perform the respective movements in order to accomplish the desired effects and avoid the unwanted ones. Thus, symbolic representations of actions are unable to specify fine-grained motions or reactions and parameterize motions to achieve the desired and prevent the unwanted effects. Symbolic approaches do not deal with the stable, controllable, and optimal execution of robot movements.

Table I and Figure 2 contrast aspects of action control easily communicated within control engineering with respective representations from an AI point of view. We feel that

Control Theory Aspects	AI Aspects
Physical measurements: <ul style="list-style-type: none"> • Position, velocity, acceleration, torque • Joint and Cartesian space • Frames on robot and objects Example: "Move the End-Effector +10cm in Z with maximum velocity and acc."	Higher abstraction: <ul style="list-style-type: none"> • Object semantics • Physical effects • Context conditions Example: "Put the clamp into the open hand from above, release on contact."

TABLE I

COMPARISON OF THE TYPICAL TERMS USED FOR EXPRESSING CONCEPTS RELATED TO ROBOT MOVEMENT FROM THE POINT OF VIEW OF CONTROL THEORY AND ARTIFICIAL INTELLIGENCE: BOTH RESEARCH COMMUNITIES SEEM TO HAVE BUILT MAINLY DISJUNCT BUT COMPLEMENTARY CONCEPTUALIZATIONS OF ACTION CONTROL.

both research fields have developed conceptualizations of robotic action control with few overlaps but which promise to complement each other rather well.

3) *Combining AI and Control Engineering Methods to Action Control*: In our approach, we couple the symbolic and the control layer directly by means of shared data structures that specify movements as first-class objects: External force disturbances of the motion controllers are directly related to safety events which are further annotated with the intents and poses of the human and robotic agents. Furthermore, the tracked body parts of the human correspond to repelling virtual environment which the motion controllers try to avoid.

In previous work [7], [8], [9] we followed a similar approach to bridge the gap between the AI and motion control modules of our robot control systems. We use a constraint-based movement specification as an interlingua for control and reasoning which is grounded in the functional properties of the objects used to perform actions. Constraint-based movement specifications can serve as fine-grained, modular, and transparent partial movement descriptions that can be interpreted by locally optimizing controllers to produce high-performance motion.

In short, we believe that the key to building AI systems for robotic agents which are aware of the effects of their movements is to have deep representations for the various parameters exposed by motion controllers. To avoid shallow integration, we start our research with shared common representations.

B. Cognition-enabled Robot Control

In our work, we follow the approach of *cognition-enabled robot control* [5]. This paradigm of building the control programs of robotic agents rests on three core principles.

Principle 1: We specify the behavior of our robotic agents in a high-level programming language which supports concurrency and reactivity – so-called *plans*. As a result, we can compactly express behavioral patterns, e.g. safety monitoring such as the one shown in algorithm 1, because our plans take advantage of language constructs for describing parallel

task execution, naming, starting, stopping, suspending, and restarting of tasks.

Algorithm 1 Compact specification of concurrent and reactive behavior in cognition-enabled control: A robotic agent shall perform a motion, while in parallel ensuring basic safety by monitoring the distance of the human agent.

```

1: (def-plan safely-move (motion human robot)
2:   (pursue
3:     (:tag motion-task
4:       (retry-after-suspension
5:         (perform robot motion)))
6:     (:tag human-tracking-task
7:       (perceive human))
8:     (:tag distance-monitoring-task
9:       (whenever (human-close-to-robot)
10:        (with-task-suspended (:motion-task)
11:          (wait-for (human-far-robot))))))

```

Principle 2: We employ reasoning mechanisms to decide on the course of action execution and action parametrization during the execution of plans. In our experience, decision making during task execution either decides on which action to perform next, or how to perform a given task. Algorithm 2 shows examples of inferring action parametrization and action activation using – among others – beliefs over the states of the human and robotic agents in lines 5 and 9, respectively. We believe that stating such control decisions as execution time reasoning tasks increases both the reusability (as programmers can program against descriptions of objects, agents, events, or actions) and reactivity of our robot control programs.

Algorithm 2 Framing control decision as reasoning tasks handled by external knowledge bases increases both the generality and reactivity of plan-based behavior specifications. Using continuously updated data, virtual knowledge bases can decide whether (line 9 and 11) or how (line 5) to perform certain actions.

```

1: (def-plan safely-move-kb (action human robot)
2:   (pursue
3:     (:tag motion-task
4:       (retry-after-suspension
5:         (perform robot (infer-params action)))
6:     (:tag human-tracking-task
7:       (perceive human))
8:     (:tag distance-monitoring
9:       (whenever (infer (is-close human robot))
10:        (with-task-suspended (:motion-task)
11:          (wait-for (infer (is-far human robot))))))

```

Principle 3: We semantically annotate the plans of our robotic agents which facilitates designing modular plans, manipulation of robot control program at run-time, and reasoning about past executions. As we do not exploit the benefits of this principle in the remainder of this paper, we refer the reader to paper [5] for a more detailed discussion.

In the subsequent chapters we try to show how the principles of cognition-enabled robot control help us design robotic agents which exhibit human-safe behavior and understanding of safety-related action aspects. Specifically, principle 1 allows us to augment safety-agnostic task execution with parallel safety-monitoring. Additionally, by following

principle 2 of formulating run-time decisions as questions to reasoning mechanisms, e.g. safety stops or choosing appropriate velocity and stiffness thresholds etc., we have to make the safety-related semantics of these choices explicit to the control system.

C. Safety-aware Cognition-enabled Control

In order to enable robotic agents to exhibit competent behavior including physical interactions with humans, we need to make robotic agents safety-aware. To this end, we propose to introduce three additional categories of first-class objects into the programming language of cognition-enabled robot control: human co-workers, robot motions, and safety-relevant events.

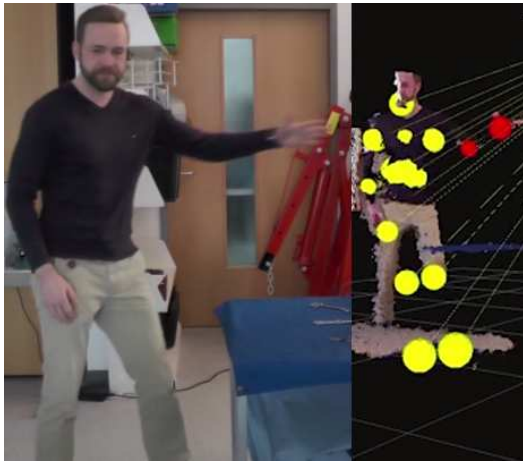


Fig. 3. A human acting in the workspace of the robot and an exemplary internal representation in the cognition-enabled plan. Body parts being so far away that the robot cannot physically interact with them are colored in yellow and close safety critical body parts in red.

Humans: The robotic agent has to continually monitor the environment to watch out for approaching humans. The robot maintains a dynamic kinematic model of the humans in its workspace where the links in the kinematic model are labeled with identifiers of the respective body parts, e.g. hands, elbow, head, and link structures such as the right arm. The body parts themselves are represented as spheres which are automatically tracked using a human pose tracking system (see Section IV for details, and Figure 3 for visualization). As a result, the robot has the means to build up the semantics of the various human body parts. During a hand-over, for instance, it is OK for the robot to approach the human hand with its own hand, while it is virtually never OK for any robot body part to approach the human head.

Robot motions: The robot motions are parameterizable symbolic descriptions with attractors, stiffnesses, and limits for speed and accelerations. Automatically-triggered safety reactions and virtual repelling objects are also part of the robot motion representations. As these match the commands accepted by the safety-aware motion controller (compare section III), the task executive of the robotic agent may directly specify how its arms shall perform movements – a prerequisite for competent behavior control.

Safety events: The safety events are descriptions of situations such as detection of a new human, a human coming close to the robot, a registered external force, or a safety reaction automatically activated by the motion controllers. More complex events can also be constructed out of other events, e.g. the human hand coming very close to the robot hand *and* registration of a slight external force in the same robot arm. Events are automatically described and sent to the cognition-enabled control system where concurrent event-directed sub-plans are triggered as reactions.

These first-class object descriptions – humans, safety events, and parameterizable motions – are not only data structures which the task executives passes to control routines for parameterizing actions or which it receives from perception modules as feedback. Additionally, they are also representations which reasoning components can combine with context information concerning objects and scenes in order to cooperate with humans more competently.

D. Software Framework

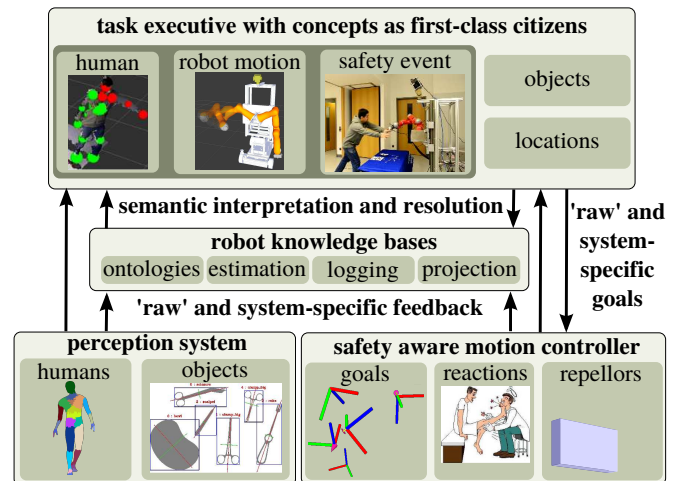


Fig. 4. Architecture of a safety-aware robotic agent: Domain-specific robot knowledge bases interpret the robot's efferent and afferent raw information, effectively creating the semantic interpretations which make up the conceptual apparatus of the task executive: First-class representations of humans, robot motions, and safety events.

A safety-aware cognition-enabled robot control system consists of four components (see Figure 4). Let us introduce them by starting at the bottom of the architecture and going up. Our cognition-enabled robot control system recruits raw feedback from both the perception system (Section IV) and the safety-aware motion controller (Section III). Domain-specific robot knowledge bases log the raw data, and offer semantic interpretation of the information via on-demand queries. Such knowledge bases are coined *virtual knowledge bases*, and we refer the reader for more details on the paradigm behind them to [10]. Every time new raw feedback is available, the task executive triggers semantic queries to fill representations, directs the flow of control accordingly, employs further reasoning to disambiguate its goal, and sends the next raw commands to the controller.

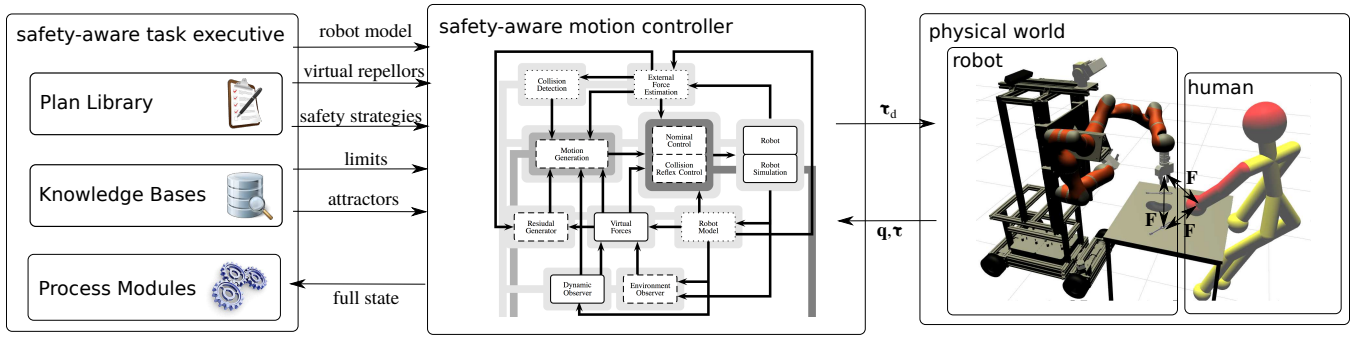


Fig. 5. Raising the level of abstraction when programming safety-aware robotic agents: Motion controllers model movements in terms of attractors, impedances, virtual repellers, safety strategies, etc. instead of using joint torques and positions. Task executives need to share these motion representations and connect them to descriptions of humans, objects, and events to predict and avoid dangerous force interactions.

III. SAFETY-AWARE MOTION CONTROL

A safety-aware robotic agent has to use its sensors and actuators such that it achieves its goals while ensuring the safety of its human co-workers. State-of-the-art robotic manipulators accept joint torque commands τ_d at high frequencies, and provide feedback in terms of measured joint \mathbf{q} and torques $\boldsymbol{\tau}$ equally fast. The motion of the links of the manipulator will cause various force interactions \mathbf{F} between body parts of the robotic agent, the human co-worker, or any object in the environment. It is those force interactions that can cause damage to humans. Here, we outline how to raise the level of abstraction from a motor-centric representation of actions to symbolic descriptions which explicitly consider humans, objects, motion, and events, as depicted in Figure 5. We believe this level of representation is essential to predict and control robot-object-human force interactions.

The safety-aware motion controller BEASTY [11] wraps joint and Cartesian impedance control schemes with various safety features. For instance, programmers can specify velocity and acceleration limits, virtual repelling walls and spheres, or safety strategies for stopping with an impedance controller or the motor brakes in case of unexpected collisions. Through its new features, this controller effectively raises the level of abstraction for motion control.

Algorithm 3 A symbolic description of an action to place an object at a location. Additional motion constraints, e.g. to avoid a human or point any dangerous parts of the object away from the human, extend it with safety commands. One can also specify which safety strategies the controller shall use in case of detected events, e.g. ignore contact events.

```

1: (def-plan safely-place-obj (obj loc human robot)
2:   (let ((place-action
3:         (an action
4:           `((type place) (acted-with ,obj) (at ,loc)
5:             (motion-constraints
6:               ((avoid ,human)
7:                 ((point away)
8:                   (acted-with (dangerous-parts ,obj))
9:                     (acted-on ,human))))
10:          (safety-strategies
11:            ((contact ignore))))))
12:     (safely-move-kb place-action human robot)))

```

However, we want to enable programmers to specify the desired behavior of robotic agents at the level depicted in Algorithm 3: Symbolic action descriptions which use concepts like objects, locations, events, or humans, and list desired or undesired effects as further motion constraints.

The safety-aware task executive CRAM [5] resolves the inherent ambiguity of action descriptions at runtime using virtual knowledge bases. On-demand disambiguation allows the robotic agent to exhibit behavior which generalizes to new execution contexts [12]. Mapping desired and undesired action effects onto specific motor commands for given contexts (and back) is one of the main components in human motor control [13]. We believe building safety-aware robotic agents will also rely on the successful development of reasoning mechanisms which performing these mappings.

The reasoning mechanisms of knowledge bases for safety-aware motion controllers have to make various safety decisions while keeping overall consistency. They shall position attractor frames and select velocity thresholds in order to place a particular object at a given location. Furthermore, they have to handle geometric constraints, e.g. add a repelling wall in front of a human to avoid touching her, or alter the attractor frame to point all dangerous object parts away from the co-worker. Finally, these knowledge bases need to specify impedance values to ensure non-violent contacts when placing an object, but also safety strategies to use in case of light contacts or strong collisions, etc. Consistently and correctly making these safety decisions relies on computational models which are subject to future research.

Knowledge bases for safety-aware robotic agents and motion controllers need to share common representations. Consider as a counter-example, the 3T architecture [14] which decoupled task executive and motion control through a software layer. In such a setup, it is impossible to build predictive models of motion controllers as their capabilities are hidden behind an interface. As a result, knowledge bases cannot connect motion commands to effects through. Hence, we think it is necessary to have deep models of motion controllers based on shared representations to build the task executives of safety-aware robotic agents.

IV. KNOWLEDGE-ENABLED PERCEPTION

Capabilities to perceive both static and rigid objects and dynamic and deformable humans is a necessary prerequisite for building safety-aware robotic agents which shall act in open real-world environments. Hence, robots need to be equipped with sensors capable of delivering high quality and high frequency data, and algorithms that can reliably interpret this data. From an image processing point of view detecting humans and objects require significantly different approaches. In numerous applications one-shot perception suffices for detecting objects and their poses in the environment. On the other hand when a human enters the robot's working area it needs to be tracked continuously. For a single perception system to handle this diversity it needs to be highly reconfigurable and adaptable to the task it is executing, be that the detection and tracking of humans or simply identifying possible locations of objects.

In previous work [15] we introduced ROBOSHERLOCK, a perception framework that builds on top of the Apache Unstructured Information Management (UIM) Architecture [16] and is designed to enable the integration of knowledge-based reasoning about the environment the robot operates in. Historically, UIM is used predominantly in natural language processing. The general idea of UIM is to have multiple expert algorithms that analyze and annotate the input data and combine these experts into processing pipelines. These properties make it an ideal framework to combine results from different sources, and allows for task dependent pipeline adaptation.

Algorithm 4 A simple perception plan instructing ROBOSHERLOCK to continuously track humans. The task executive binds the stream of percepts to a fluent which other plans can use and react to.

```

1: (def-plan human-tracking
2:   (let ((human-fluent (make-fluent)))
3:     (perform (an action
4:              '((type track) (a human)
5:                (bind-fluent ,human-fluent))))
6:   human-fluent))

```

Regarding safety-aware robotic agents there is a need for continuous tracking of humans in the workspace shared by the co-workers. The task executive of the robot will use ROBOSHERLOCK to get the information about its environment that is needed to safely interact with humans. For instance, everytime the robotic agent needs to know about close humans it can either on-demand query the perception system for a specific frame as stated in [15] or instruct ROBOSHERLOCK to continuously inform it about perceived humans (see Algorithm 4). In streaming mode, the task executive binds the incoming percepts to fluents which other plans can react to.

If ROBOSHERLOCK can perceive humans one might want to extend its capabilities to also estimate whether humans are about to perform specific actions. Describing a perceptual event in which the human is reaching for the same surgical instrument the robot wants to currently grasp is possible

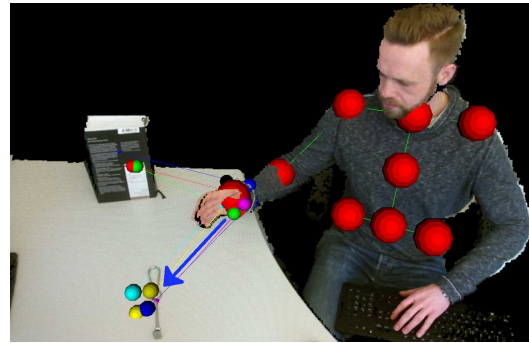


Fig. 6. The intention analyzer states that the human most probably is about to reach for the rake.

with the first-class concepts of safety-aware robotic agents. Algorithm 5 shows an abstract version of a description of such a perceptual action. Figure 6 depicts the output of ROBOSHERLOCK for this query with object restricted to surgical instruments of type rake.

Algorithm 5 An action description to detect an event in which a given human tries to grasp a given object.

```

1: (an action '((type detect)
2:             (event , (an action '((type grasp)
3:                                   (actor ,human)
4:                                   (acted-on ,obj))))))

```

Or, the robotic agent was asked to hand over the scalpel. Now, its task executive can query the perception system to perceive the blade of a scalpel in order to grasp it at the correct point and hand it over such that the handle of the scalpel is oriented towards the human:

Algorithm 6 An action to perceive the blade of a scalpel.

```

1: (an action
2:   '((type perceive)
3:     (acted-on , (an object-part
4:                 '((type blade)
5:                   (part-of , (an object
6:                               '((type scalpel))))))))))

```

Querying ROBOSHERLOCK like this adapts the basic pipeline such that the necessary experts will be added and the answer of ROBOSHERLOCK will contain the information that were asked for. This adaption can be either permanent or only for next frame that is processed.

V. EVALUATION

A. Prototypical Safety-Aware Robotic Agents

We recorded three real-world interaction experiments with a human co-worker and a leading-edge mobile manipulation platform to show the power of the concepts of safety-aware cognition-enabled control.

In the first experiment, the robotic agent performs repetitive free-space motions while ensuring the safety of the human. In case the human gets too close the robot will suspend its motion. Furthermore, it will react to strong

collisions with a stop and subsequently switch into gravity compensation mode. You can find a recording of this experiment at <http://youtu.be/OYiLTJ8YCPc>. The task executive will restart the main program once neither the robot arm reports a collision nor the perception system detects a human intrusion.

The second experiment shows the capabilities of the robotic agent to perceive, pick and place surgical instruments from a table. The task executive is instructed to place all instruments which it perceives that are not bowls into the bowl. Additionally, the safety features demonstrated in the first experiment are also active. As a result it is possible for a human co-worker to step into the workspace of the robot, interrupt it, and re-arrange the tools. We shared a recording of this experiment at <http://youtu.be/M5jwRmjK4ZM>.

Finally, we performed a similar experiment to the second one with a later prototype of our safety-aware robotic agent (see video <http://youtu.be/2SIWTH8bwU0>). Again, the robot shall place all surgical instruments into the bowl while ensuring the safety of the human co-worker. In addition to performing its task and complying with two co-workers entering and re-arranging its workspace, the task executive logs its symbolic and sub-symbolic belief state into a knowledge base for later reconstruction and analysis. This allows developers and other software agents to inspect and debug the behavior of the safety-aware robotic agent.

B. Open Research

While performing the final evaluation experiment we recorded extensive log data to allow reconstruction of the experimental run. Using our recently developed and presented online knowledge processing framework OPENEASE [17] we share this data with the general public. We invite you, the readers, to visit our OPENEASE website at www.open-ease.org to get an even better impression of the quality and quantity of representations our safety-aware robotic agents employ and which reasoning tasks these support.



Fig. 7. The OPENEASE [17] online knowledge processing system allows reviewers and readers to independently analyse the data logged in our experiments. Semantic queries using the descriptions of objects, humans, motions, and events show how safety-aware robotic agents could make sense of their low-level percepts using these concepts.

Once you're on the OPENEASE website, please register

and login using your username and password. On the subsequent screen, please open the interactive environment by selecting the newest knowrob daemon. Load the experiment associated with this paper by selecting "SAPHARI/Review-2015" on the top right, and clicking on the link "Knowledge Base" on the top left.

The following interface holds a list of sample queries in natural language on the left. Once you click on a query, OPENEASE produces the Prolog equivalent of each query in the query dialog. Instruct the system to evaluate the query by pressing the *return* key. To get a first impression of the sample queries and the logged data, please evaluate the instructions in the suggested order. Note, the first queries load and initialize the database. You can reset your session by logging out and into OPENEASE. Feel free to explore the analysis tool and discover details we have not outlined in this paper. If you experience technical difficulties using the system with your browser of choice, please consider trying it with a recent version of Firefox.

VI. RELATED WORK

Kulic and Croft present early work on safe human robot interaction in [18]. The basic idea is to introduce a function which quantifies the joint human-robot configuration w.r.t. the potential of hurting the human. A planner employs this danger function to minimize the risk of the interaction between human and robot. In subsequent work, the authors show how this approach translates to building real-time motion controllers by essentially integrating the danger function into the potential function which the controller shall minimize [19]. Explicit and semantically rich representations of the human, the task at hand, the objects both human and robot are handling or desired physical interaction were outside of the scope of [18], [19].

Alami et. al. present another framework for planning of safe human-robot interaction both for navigation [20] and manipulation [21]. They coin the term of a *human aware planner* which "must not only elaborate safe robot paths, but also plan good, socially acceptable and legible paths" [20]. Simplified speaking, their human aware agents reason whether regions of space are accessible or visible to human co-workers and also have representations for human comfort when handing over an object. They introduce and use the powerful concept of *perspective taking* to build their planning systems. Direct physical interactions, however, are neither treated by the symbolic reasoner nor by the motion controllers.

Ragaglia et. al. [22] present a state-of-the-art safety-aware industrial robotic agent architecture which also integrates safety-aware motion control software with visual perception of humans and objects. However, as they employ a high-mass and non-compliant robot arm which may exert dangerous forces their executive does not consider preparing for or reacting to physical interaction with human agents. Instead, they extend standard motion controllers with algorithms which evade human contact, and build an agent system

which context-sensitively chooses the appropriate avoidance strategy, similar in approach to [19].

Eventually safety-aware robotic agents need access to knowledge bases which provide the necessary background information over the characteristics and consequences of physical interactions between humans and robots. Behrens and Elkmann [23] argue convincingly that "only comprehensive collision tests with live test subjects lead to verified limit values" for human-robot collisions, and present test methodology, similar in fashion to our work in [24].

VII. CONCLUSION

In this paper we have extended the conceptual framework of cognition-enabled control to the domain of safety-aware robotic agents. The framework enables the task executive of the robot to consider both the task as well as the safety of human co-workers in its control decisions. The main difference to typical AI-based architectures such as the 3T architecture [14] is that the control layer is not decoupled from the high-level control through an additional software layer. Rather, key concepts for specifying safety-aware control, including a model of the human co-worker, the safety-relevant events, and the robots movements are explicitly represented in the high-level plan language. This enables the robot to adapt its movements in a context-directed manner through symbolic reasoning and thereby make full use of the robot's low-level motion and perception capabilities. We have shown the power of the safety-aware control approach in a real-world scenario with a leading-edge autonomous manipulation platform. Furthermore, we have shared recordings of our experiments through the online knowledge processing system OPENEASE.

ACKNOWLEDGEMENTS

This work was supported in part by the EU FP7 Projects *RoboHow* (Grant Agreement Number 288533) and *SAPHARI* (Grant Agreement Number 287513).

REFERENCES

- [1] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3356–3363.
- [2] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.
- [3] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by Kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, June 2011, pp. 15–22.
- [4] K. Buys, C. Cagniart, A. Baksheev, T. D. Laet, J. D. Schutter, and C. Pantofaru, "An adaptable system for rgb-d based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1047320313000515>
- [5] M. Beetz, D. Jain, L. Möslechner, M. Tenorth, L. Kunze, N. Blodow, and D. Pangercic, "Cognition-enabled autonomous robot control for the realization of home chore task intelligence," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2454–2471, 2012.
- [6] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Rob. Res.*, vol. 26, no. 5, pp. 433–455, 2007.
- [7] I. Kresse and M. Beetz, "Movement-aware action control – integrating symbolic and control-theoretic action execution," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012, pp. 3245–3251.
- [8] G. Bartels, I. Kresse, and M. Beetz, "Constraint-based movement representation grounded in geometric features," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Atlanta, Georgia, USA, October 15–17 2013.
- [9] M. Tenorth, G. Bartels, and M. Beetz, "Knowledge-based specification of robot motions," in *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2014.
- [10] M. Tenorth and M. Beetz, "KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566 – 590, April 2013.
- [11] S. Parusel, S. Haddadin, and A. Albu-Schaffer, "Modular state-based behavior control for safe human-robot interaction: A lightweight control architecture for a lightweight robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4298–4305.
- [12] D. McDermott, "Robot Planning," *AI Magazine*, vol. 13, no. 2, pp. 55–79, 1992.
- [13] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Current Opinion in Neurobiology*, 2006.
- [14] P. Bonasso, J. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack, "Experiences with an Architecture for Intelligent, Reactive Agents," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, no. 1, 1997.
- [15] M. Beetz, F. Balint-Benczedi, N. Blodow, D. Nyga, T. Wiedemeyer, and Z.-C. Marton, "RoboSherlock: Unstructured Information Processing for Robot Perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, best Service Robotics Paper Award, Finalist for the Best Conference Paper Award.
- [16] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefel, and C. Welty, "Building Watson: An overview of the DeepQA project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010. [Online]. Available: <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303>
- [17] M. Beetz, M. Tenorth, and J. Winkler, "Open-EASE – a knowledge processing service for robots and robotics/ai researchers," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, finalist for the Best Conference Paper Award and Best Cognitive Robotics Paper Award.
- [18] D. Kulić and E. A. Croft, "Safe planning for human-robot interaction," *Journal of Robotic Systems*, vol. 22, no. 7, pp. 383–396, 2005.
- [19] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007.
- [20] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM, 2005, pp. 81–85.
- [21] E. A. Sisbot, L. F. Marin-Urias, X. Broquere, D. Sidobre, and R. Alami, "Synthesizing robot motions adapted to human presence," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 329–343, 2010.
- [22] M. Ragaglia, L. Bascetta, P. Rocco, and A. M. Zanchettin, "Integration of perception, control and injury knowledge for safe human-robot interaction," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1196–1202.
- [23] R. Behrens and N. Elkmann, "Study on meaningful and verified thresholds for minimizing the consequences of human-robot collisions," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3378–3383.
- [24] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer, "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.