BlueSky: Combining Task Planning and Activity-Centric Access Control for Assistive Humanoid Robots

Saskia Bayreuther
saskia.bayreuther@kit.edu
KIT
Karlsruhe, Germany

Florian Jacob
florian.jacob@kit.edu
KIT
Karlsruhe, Germany

Markus Grotz
markus.grotz@kit.edu
KIT
Karlsruhe, Germany

Rainer Kartmann
rainer.kartmann@kit.edu
KIT
Karlsruhe, Germany

Fabian Peller-Konrad
fabian.peller-konrad@kit.edu
KIT
Karlsruhe, Germany

Fabian Paus
fabian.paus@kit.edu
KIT
Karlsruhe, Germany

Hannes Hartenstein
hannes.hartenstein@kit.edu
KIT
Karlsruhe, Germany

Tamim Asfour
asfour@kit.edu
KIT
Karlsruhe, Germany

ABSTRACT

In the not too distant future, assistive humanoid robots will provide versatile assistance for coping with everyday life. In their interactions with humans, not only safety, but also security and privacy issues need to be considered. In this Blue Sky paper, we therefore argue that it is time to bring task planning and execution as a well-established field of robotics with access and usage control in the field of security and privacy closer together. In particular, the recently proposed activity-based view on access and usage control [8] provides a promising approach to bridge the gap between these two perspectives. We argue that humanoid robots provide for specific challenges due to their task-universality and their use in both, private and public spaces. Furthermore, they are socially connected to various parties and require policy creation at runtime due to learning. We contribute first attempts on the architecture and enforcement layer as well as on joint modeling, and discuss challenges and a research roadmap also for the policy and objectives layer. We conclude that the underlying combination of decentralized systems' and smart environments' research aspects provides for a rich source of challenges that need to be addressed on the road to deployment.

CCS CONCEPTS

• **Security and privacy → Access control**; **Network security**.

KEYWORDS

access control, usage control, robot task planning, humanoid assistance robotics

ACM Reference Format:
Saskia Bayreuther, Florian Jacob, Markus Grotz, Rainer Kartmann, Fabian Peller-Konrad, Fabian Paus, Hannes Hartenstein, and Tamim Asfour. 2022. BlueSky: Combining Task Planning and Activity-Centric Access Control for Assistive Humanoid Robots. In *Proceedings of the 27th ACM Symposium on Access Control Models and Technologies (SACMAT) (SACMAT '22), June 8–10, 2022, New York, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3532105.3535018

SACMAT '22, June 8–10, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9357-7/22/06…$15.00
https://doi.org/10.1145/3532105.3535018

1 INTRODUCTION

Technological advances let the deployment of humanoid robots as assistive service robots, in particular for healthcare tasks [19], appear realistic. However, as one can observe in the field of autonomous driving, having the required capabilities to perform a task is a necessary but not a sufficient condition when it comes to the question of what a robot is allowed to do. The need for "Formalizing and Guaranteeing Human-Robot Interaction" [14] has been prominently articulated recently. In this Blue Sky paper, we put forward the thesis that it is time to bring policies, models, and architecture used in robot task planning and in access and usage control closer together, particularly with the advent of the recently-proposed activity-centric view on access and usage control [8] in mind. A robot performs *task planning* "to plan a sequence of high-level actions that allows the robot to perform a given task" [4]. Robot task planning, thus, is the step between a robot observing its environment and manipulating its environment. Naturally, robots have to comply with safety regulations to not harm their environment, especially humans. When dealing with humans and information about humans, however, robot task planning also needs to be concerned with privacy and security aspects. Thus, merging robot task planning with access and usage control promises an integrated "safety-security-privacy-by-design." Only by merging robot task planning (what the robot is able to do) with access control (what the robot is authorized to do) in the robot architecture by design, can one achieve enforcement of the complex security and privacy policies required for humanoid assistive robotics. We underline this hypothesis using the four challenges **C1–C4** below.

In this paper, we will solely focus on versatile assistance robotics for supporting humans in everyday life. Let us assume the following scenario: humanoid assistive robots are used in an assisted living home where people of every age live and are assisted by care workers. The robots should be deployed in a way that aids the residents in their daily life and lightens the care workers' workload. The living home consists of private apartments and common space such as a kitchen, dining and living rooms, and a balcony. The robots can move freely in common space areas where no physical thresholds

have to be passed. The robots can interact with humans via different channels, mainly natural speech commands, hand gesture and display. The robots' duty is to listen to and recognize commands given by humans, identify and authenticate the person and compile and execute the task based on the command in compliance with enforced security and privacy policies.

The sketched scenario points to major features and challenges:

C1 Humanoid robots are *task-universal*, i.e., they can basically do what a human being can do. The corresponding challenge is that a humanoid robot is not "restricted-by-design," but instead requires a specification and implementation of what it is allowed to do with the necessary complexity to match its versatility.

C2 Humanoid robots will *live in a private space* of a person, but will also move to public spaces. The corresponding challenge for privacy can be seen as equivalent to letting a stranger in one's own home — and the stranger is more powerful than Alexa or other cloud-based voice-controlled personal assistant systems as indicated in challenge C1.

C3 Humanoid robots act in a dynamic, *socially connected* environment. They do not only act for the assisted person, but potentially also for other care takers, a robot operator, an operator of the assisted living home, one or several medical doctors, and family members, to name a few. While a robot's environment is dynamically manipulated by those persons as well as by the robot itself, the robot has to fulfill various obligations and conditions. The corresponding challenge lies in the amplification of security and privacy requirements that have to be fulfilled through policy enforcement.

C4 Policy creation is *required at runtime and involves multiple parties with different interests*, i.e., policy conflicts have to be resolved between the above mentioned parties, and policy creation might be needed while a robot is performing a task. Thus, the corresponding challenge points to agreement processes for policy creation, i.e., the administrative model.

The claim of task universality of future humanoid robots can be supported by the fact that these robots are developed with a high number of the degrees of freedom to resemble human versatility in executing a wide variety of tasks. As an example, the humanoid robot ARMAR-6 [2] has 28 degrees of freedom to support human-robot collaborative tasks. With this flexibility, humanoid robots can also actively manipulate their environment to achieve given task goals and objectives, which needs to be reflected in planning and access control for such robots.

Being part of both private and public spaces together with the social connectedness of humanoid robots comes with an attack surface for safety, security, and privacy. A corresponding challenge can be illustrated using the example of disease management: administering medication or instructing physical exercises obviously combine safety, security as well as privacy implications. Therefore, without a combined treatment of access/usage control and robot task planning, no appropriate system architecture is possible.

Since it is unlikely that all policies are predefined, an ad-hoc governance and agreement procedure for policies is required. Unlike traditional social media where users can define policies which are enforced by a centralized entity, we are in need of distributed or decentralized policy creation and management [24]. This aspect might be further complicated in a scenario with multiple humanoid assistive robots interacting with humans or with each other.

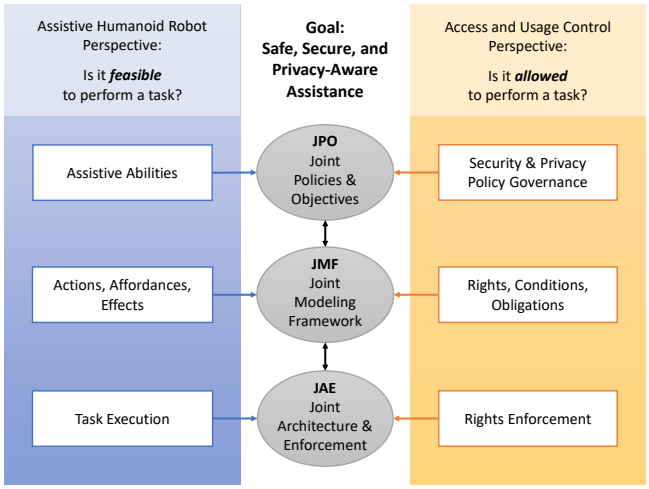

Figure 1: The guiding questions of both, assistive humanoid robots as well as access and usage control engineering, need to be addressed by a joint effort on the policies & objectives, modeling, and architecture & enforcement layers. While the layerist approach follows standard models OM-AM [21] and PEI [22] (the implementation layer is not shown in our illustration), there might be unique interactions and dependencies within and between the layers due to robots' abilities to learn and to actively manipulate their environment.

As illustrated in Figure 1, there is a natural dependency and tension between the aspects of what a humanoid robot is capable to do and of what it is allowed to do. To deal appropriately with this tension, a joint approach on the layers of policy & objectives, modeling, as well enforcement & architecture appears to be required. The layers Objectives-Models-Architecture-Mechanisms proposed in [21] as well as Policy-Enforcement-Implementation in [22] are valid also in an assistive humanoid robotics scenario. However, it is less clear how and to which degree the "joining" could be done and the interactions between the layers can be strictly clarified. These issues represent the "leitmotif" of this Blue Sky paper.

In this paper, we approach the topic based on the structure of Figure 1 in a bottom-up fashion as follows:

- **JAE:** On the basis of a system model for a humanoid robot, we first indicate on which level or layer a combined treatment of access and usage control as well as robot control is required and appropriate. The analysis includes a delineation between lower level robot control and robot task planning. Furthermore, we illustrate the distributedness of policy information, decision, and enforcement points in an assisted living scenario [Section 2].
- **JMF:** We then show exemplarily how robot task planning and activity-centric access and usage control models can be effectively combined, but also come with specific requirements on modeling aspects [Section 3].

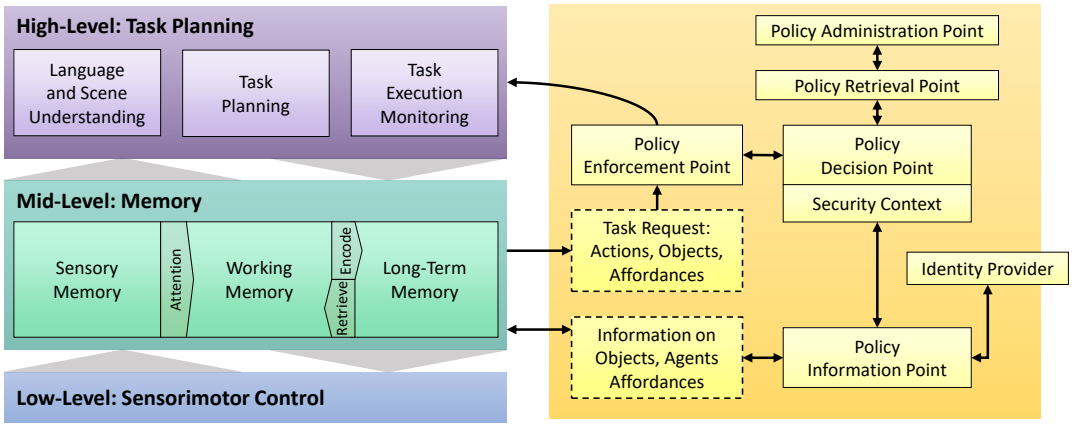

Figure 2: Joining the cognitive architecture of a humanoid robot with access and usage control. While the left side represents the architecture of ArmarX [3, 25], the right side should be considered a first implementation effort to join access and usage control modules. When the robot receives a command, first task planning issues a task request to the access and usage control via the memory. The Policy Information Point has access to the robot's memory to inform the Policy Decision Point about the necessary objects, agents and affordances. The Policy Enforcement Point signals access control decisions to the task planning and execution monitoring continuously. The Security Context represents a module for usage control tasks, i.e., attribute mutability and decision continuity.

- **JAE + JMF + JPO:** We comment on the practicality of the modeling implementation as well as on decentralized policy governance and distributed enforcement, and outline a research agenda to make privacy-aware humanoid robots in assisted living homes a reality [Section 4].

Related work in the fields of smart environments, particularly smart homes and automotive systems, are presented in Section 5. We conclude in Section 6.

2 TOWARDS JOINT ARCHITECTURE & ENFORCEMENT

We first describe the cognitive architecture of a humanoid robot and then indicate and discuss where access and usage control components relate to the cognitive architecture, see Figure 2. The joint architecture outlined above is implemented in a prototype as a first step, see Figure 3.

A humanoid robot can perceive the environment with different sensors, interact with humans using speech and perform activities such as grasping and placing objects, pouring juice into a cup, cleaning or setting up the table for dinner. Given a task specified by a natural language command given by a human, the robot has understood the command and generates a plan as sequence of actions to achieve the goal of the task. Such a cognitive ability requires integrating natural language understanding, symbolic planning, plan execution monitoring and reasoning, comprehensive scene perception and interpretation, and sophisticated sensorimotor control. In [27], we showed how we can endow humanoid robots with such abilities. To this end, we proposed a cognitive control architecture that integrates (i) semantic symbolic representations in natural language, planning and reasoning and (ii) sensorimotor continuous representations needed for execution and control, see also [3]. The architecture, see Figure 2 (left), consists of three layers:

(1) The *sensorimotor* low-level. This level incorporates a hardware abstraction component and the ArmarX statecharts (see [28]) for the execution of sensorimotor skills. The statecharts combine in a task-specific way sensorimotor data of the low-level and processed data of the memory layer to implement higher level skills such as grasp, lift, open, wipe, etc.

(2) The *memory system* mid-level. This level consists mainly of the memory structure with three types of memory: The sensory memory, the working memory and long-term memory. The memory system acts as mediator between the low- and high-level and is equipped with processing capabilities like object recognition, object detection, hand tracking, self-localization, etc. The sensory memory is a repository for incoming raw sensory data provided by the different sensors. The working memory is a volatile memory in which all available information about the current state of the world such as objects and their position, humans and their actions are stored. The long-term memory combines (i) prior knowledge about object and environment models, human body models, pre-defined grasping information, as well as semantic knowledge and facts about the world and (ii) an episodic memory that contains episodes of experienced events occurring in a given place at a given time.

(3) The *task planning* high level. This level is concerned with task planning and symbolic reasoning to facilitate natural communication with the human for solving complex tasks. The major components are language and scene understanding, task planning and reasoning, and task execution monitoring.

Compared to other cognitive architectures [5, 9, 11, 26, 30], we adopt a developmental approach to using a hybrid cognitive architecture where functional modules can be implemented using different techniques but respecting the overall constraints imposed by the architecture.

Of those three layers, the memory layer is of especially high interest from the perspective of security and privacy, since the

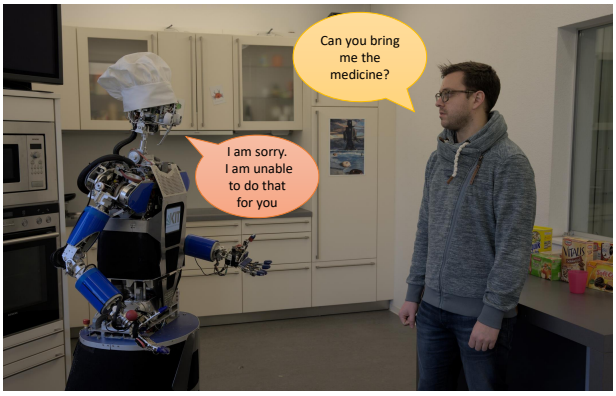

Figure 3: An illustration of the scenario. The humanoid robot ARMAR-III checks the request by the user and denies it since the user lacks the required permissions for the task. A first implementation is presented in a real robot experiment, see the video available at https://youtu.be/xs0te3HPoHc.

memory contains prior information about the residents or the access levels as well as learned knowledge from the robot's experiences. The memory as mediator between sensorimotor control and task planning serves as the place in which new semantic knowledge is learned through abstractions from sensorimotor data and symbolic plans for a given task are parameterized according to the state of the world. Introducing such a memory in the architecture represents a promising approach towards solving the so-called *signal-to-symbol gap* in technical cognitive systems [15].

The right side of Figure 2 sketches how access and usage control components interact with the robot's cognitive architecture. First, data stored by the robot can be used for access control, but can also be considered as a resource that needs protection by access control. We assume that data at the low level of the architecture are not directly accessible but only information of the mid and high level are accessible. This assumption, of course, is "by design" and must be carefully checked for side channels. Data access itself can be considered as a specific task and, thus, be subsumed under the general notion of a task.

Let us provide a walk-through of Figure 2 of an explicit request by a person who needs assistance, and a corresponding analysis of some challenges:

- The assistance request will typically be done via speech or gestures. The request can be considered as an ad-hoc policy that allows the robot to perform the requested task. However, typically this ad-hoc policy cannot be directly given to the Policy Administration Point (PAP) since a task planning is first needed to determine the required actions for the task at hand and their parameters. This raises the question: how much task planning is done before access and usage control is checked?
- When a task request is given to the Policy Enforcement Point (PEP), all data and information the robot stores in the memory could be considered as attributes that can be used for checking corresponding authorizations, obligations, and conditions of that request. The robot could even actively gain additional information by either performing exploratory actions or asking the human for clarifications. What information is handed over with the request, what information is handled via the Policy Information Point (PIP)?
- On one hand, the Policy Decision Point (PDP) with its Security Context modules keeps track of attribute mutability and decision continuity. On the other hand, the execution of the task by the robot leads to a change in the environment. How often are access and usage control checks required? Which events trigger an access and usage control reevaluation?
- Policies are administered not only by the person who is assisted by the robot, but by several other parties. The robot itself interacts with various other identities in a smart living environment. Information on other identities and the environment might be stored outside of the robot. How are the various interactions of this decentralized system designed?

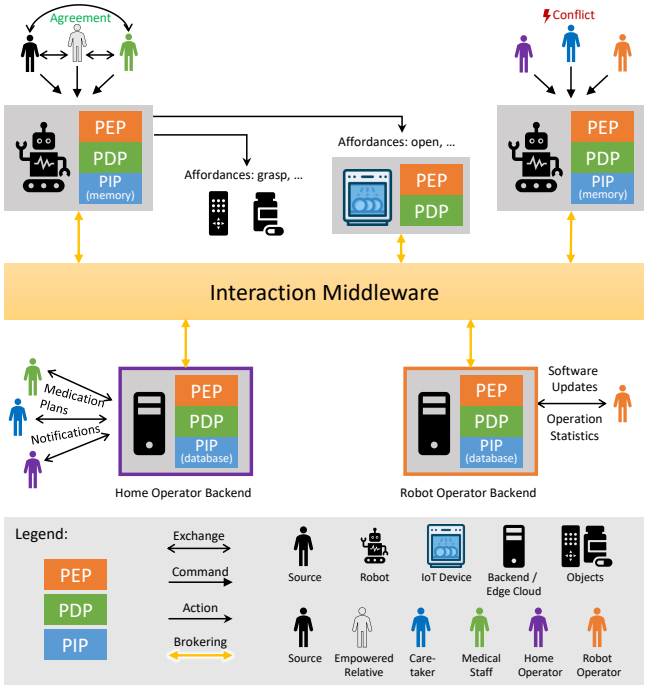

Figure 4: The decentralized nature of assisted living creates the need for agreement and conflict resolution mechanisms as well as an interaction middleware: not only policy enforcement/decision/information points (PEP, PDP, PIP) but also policies are governed by a multi-party system.

As illustrated in Figure 4, various policy points are distributed over various real world entities. In particular, there might be several robots interacting with each other. We will further discuss the interactions and the possibly decentralized nature of at least PDP, PIP, and PAP in Section 4. In the following section, we explore options for a joint modeling of task planning and access/usage control as this appears to be of fundamental importance for dealing with the "concurrency" of these two control perspectives.

3 TOWARDS A JOINT MODELING FRAMEWORK

Roboticists typically use action and/or planning languages, see e.g. [12], while security engineers make use of access and usage control meta-models. The challenge lies in finding common grounds and combination of the respective views. We start with how tasks can be modeled and what requirements exist for modeling assistive humanoid robot scenarios. Then, we devote two subsections describing attempts to check the applicability of activity-centric access control as well as of classical usage control models, respectively.

3.1 Basic Task Modeling and Requirements

When a robot plans a task, it has to generate a sequence of robot actions that transfer the current world state into a goal state that fulfills the goal of the task. A task is the result of processing a *command* given by a *source* (a human) to the robot calling for some form of assistance. For example, a human could give the command "Bring me the bottle from the table." The robot considers the source and itself as *subjects*. The robot translates the command into a task, which is decomposed into different *activities* that are executed consecutively. A task can also consist of a single activity. An activity consists of an *action* on *objects*. We consider physical objects like tables, bottles and persons as well as virtual objects like health records, personal information and images. In the example task, the actions include "move to," "grasp," and "lift," the objects include the "bottle" and the "table," and one of the activities of the above example is "move to the table." Formally, a task $T = [T_1, \ldots, T_n]$ consists of a sequence of activities $T_i = [A_i, O_{i1}, \ldots, O_{im}]$, each of which consists of an action A_i involving one or several objects O_{ij}, respectively. For example, the command "Bring me the bottle from the table" from the source "person" can be expressed as the following task:

$$T = [[\texttt{move\_to, table}], [\texttt{grasp, bottle}], [\texttt{move\_to, person}], [\texttt{hand\_over, bottle, person}]] \quad (1)$$

This form can easily be used for access/usage control. Here, we can derive two important requirements for modeling, which can be seen even in our simplistic example:

- *Sets of subjects and delegation* need to be explicitly considered since the robot typically, but not necessarily, acts as a delegate, and access rights might depend likewise on task giver and robot.
- *Different granularities of affordances and access rights* need to be considered. Affordances describe functional object properties and interaction possibility of an agent with an object [1, 7], e.g., the bottle can be grasped by the robot. But access rights might depend on the content of the bottle or on how often the bottle was given already to the task giver.

Each object can be associated with several affordances, i.e. interaction possibilities with this object. For example, a bottle affords filling, pouring, grasping, drinking and a health record of a resident affords reading, writing, displaying, and verbalizing. The robot can execute the activity "grasp a bottle" if and only if the bottle has the affordance "grasping."

There are further interacting aspects of a joint modeling:

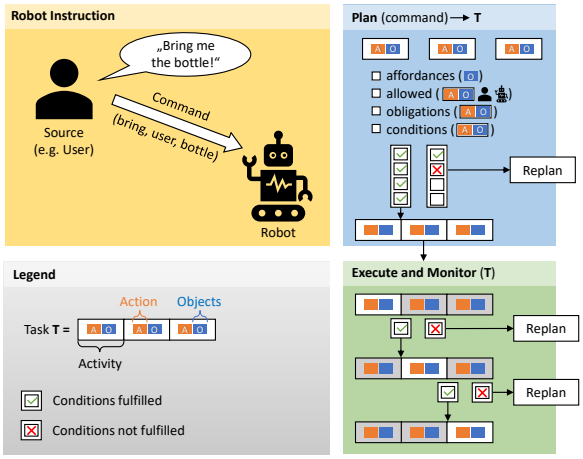

Figure 5: Illustration of interaction between task planning and access as well as usage control. A task is initiated by a command. Planning a task requires checking access control requirements such as affordances, authorizations ("allowed"), conditions and obligations. While executing the task the robot has to constantly check for valid conditions.

- *Conditions and obligations as tasks*: a condition that "a human caretaker needs to be in the room" might need to be interpreted as the task of checking whether a human caretaker is actually in the room. Similarly, an obligation might lead to a task for the robot.
- *Learning*: the robot can continuously learn and add attributes and affordances to objects at run time. For instance, the vaccination status of a person can be updated by reading a QR code of a vaccination certificate. In case of an unknown object, the robot might recognize its affordances and whether the robot can (physically or digitally) access the object.
- *Cooperative abilities*: opportunities and risks of "collusion" need to be explicitly taken into account.
- *Privacy requirements*: tasks performed by the robot can leak private information of the assisted persons. Thus, sensitive information needs to be labeled and might influence task execution.

While this list is most likely not complete, the requirements are quite extensive in order to be able to check for safety, security, privacy, and assistance functionality at design- and runtime.

The basic setup is illustrated in Figure 5: a command of a source, e.g., the assisted person, initiates task planning (which can be considered a task itself) on the robot. If a suitable plan for the task is found, the robot will start to execute the task. Of course, during execution, changes of conditions need to be checked or detected and replanning might be required. Thus, task execution monitoring has to interact very closely with access and usage control. In the following two subsections, we sketch how access and usage control policies can be expressed on the level of an activity of a task, first using activity-centric access control, second with usage control. However, replanning, i.e., the feedback from access or usage control to task planning, is not yet considered.

3.2 An Attempt based on Activity-Centric Access Control

Activity-centric access control (ACAC) [8] has been drafted for smart collaborative ecosystems with multiple smart devices, like sensors, motors and lights, that share information about their current state, i.e. their current activity. In ACAC, the execution of an activity depends on other activities and hence "access" to an activity can be restricted or denied if certain conditions are not fulfilled. State transitions in robot task planning, therefore, can be mapped to ACAC. However, we are interested to see how some basic requirements of the previous subsection can be addressed.

An *activity* control expression consists of an *operation* that is requested by a *source* on an *object* [8, Section 4],

```
Object: object
  ⟨ (operation, subjects, activity)
    (pre_conditions)
    (current_conditions)
    (resulting_conditions)
    (contextual_conditions) ⟩
```

where pre_conditions, current_conditions, and resulting_conditions can be expressed as ({pre/current/new}_state, object, subject). An activity contains an action which is executed if conditions are fulfilled. Activities and their transitions are controlled by their relation to other activities and to certain environmental conditions. The notion of an operation corresponds to an action in our terminology of Section 3.1, while the notion of activity is maintained.

Already at this point we want to note that, as opposed to the original ACAC approach, we see the need to allow multiple subjects to be associated with an activity, i.e., a set $S = \{S_1, \ldots, S_n\}$ of subjects S_i instead of only a single one, as outlined in the previous subsection. Furthermore, we see the need to parameterize activities to allow handling of multiple different objects: for example, a moving activity needs as object where the robot should move to. We now first demonstrate how ACAC can be applied to robot task planning and, then, discuss some challenges we discovered.

Example Task 1: Loading the dishwasher. The robot has the task to operate the dishwasher which consists of the activities to open, load, close and turn on the dishwasher. For each activity, the dishwasher must be in a certain state to allow the activity. The dishwasher can only be loaded if the dishwasher was previously open and empty (i.e. "has enough space to load the current dish"). Additionally, the dishwasher can only be loaded with objects that afford *dishwashing*. Also, the dishwasher should offer the affordance of *grasping* such that it can be opened and closed. We assume that the robot and the source are privileged to operate the dishwasher. The resulting condition is the increased number of dishes in the dishwasher.

The example starts with a human expressing a command to the robot: "Load the dishwasher with the dishes on the table." Assume the current state of the dishwasher is empty and closed. The robot observes this state, recognizes one dish on the table and processes the command into the task:

$$T = [[\texttt{move\_to, dishwasher}], [\texttt{open, dishwasher}], [\texttt{move\_to, table}], [\texttt{grasp, dish}], [\texttt{move\_to, dishwasher}], [\texttt{load, dish, dishwasher}], [\texttt{close, dishwasher}]]$$

We have a source, which is the human, the robot as a subject executing the task as well as the objects `robot`, `dish`, `dishwasher`, and `table`. Next, we demonstrate the policies that are associated with the activities in the task:

```
Object: robot, object
  ⟨ (move_to, robot, moving)
    (move_to ∈ affordances(object)) ⟩
```

The moving activity is parameterized with an object to which the robot should move. To execute the activity, the condition `move_to` ∈ `affordances(object)` needs to hold. Similarly, we can define policies for other activities, for example, as follows:

```
Object: robot, object
  ⟨ (grasp, robot, grasping)
    (grasp ∈ affordances(object)) ⟩

Object: robot, dishwasher
  ⟨ (open, robot, opening)
    (pre_closed, dishwasher, ANY)
    (new_open, dishwasher, ANY)
    (grasp ∈ affordances(dishwasher)) ⟩

Object: robot, dishwasher, object
  ⟨ (load, robot, loading)
    (pre_open, dishwasher, ANY)
    (pre_empty, dishwasher, ANY)
    (new_dishes + 1, dishwasher, ANY)
    (dishwash ∈ affordances(object)) ⟩
```

The keyword ANY refers to any activity. Before we comment on this type of ACAC modeling, let us first have a look at an example with a physical interaction between robot and human as well as a potentially changing environment.

Example Task 2: Bringing a bottle from someone's personal room. The robot receives the task to bring a person a bottle from their personal room. Assume that the person is the source of the task. The policy defines that the robot may only be in a personal room if the resident gave their permission and they are present as well.

The task "Bring me the bottle from my room" is formalized as follows, involving the source (`person`) and the objects `room`, `bottle`, and `person`:

$$T = [[\texttt{enter, room}], [\texttt{move\_to, bottle}], [\texttt{grasp, bottle}], [\texttt{move\_to, person}], [\texttt{hand\_over, bottle, person}]] \quad (2)$$

The entering and handing over activities can be defined as follows:

```
Object: robot, room, source
  ⟨ (enter, robot, entering)
    (enter ∈ affordances(room)
     ∧ is_resident(source, room)
     ∧ is_inside(resident(room), room)) ⟩

Object: robot, object_to_hand_over, target
  ⟨ (hand_over, robot, handing_over)
    (grasp ∈ affordances(object_to_hand_over)
     ∧ receive ∈ affordances(target)) ⟩
```

The `hand_over` activity has two associated objects: the object to hand over and the target, e.g. the person to receive the object.

These two examples illustrate that, in principle, the affordances concept of robot task planning can be modeled using ACAC. However, we also note some challenges:

- Modeling of interactions can become quite cumbersome when many objects are involved in the interaction.

- Expressing rights and policies of sources can only be done via conditions.
- While the robot executes an activity (e.g., moving to the bottle), its surroundings may constantly change, which can cause the initial checks to become outdated. Thus, the robot has to constantly check contextual conditions while executing the activity. This requirement could be encoded as a current_condition, but it remains unclear how frequently the robot needs to reevaluate these checks.
- Affordances and permissions cannot be completely separated. For example, if an object is alcoholic and the source of the task is a child, the robot, acting on behalf of the source, should not have the affordance of grasping this object. If the source is a care worker, the robot is allowed to execute the task. Before starting to execute the task, the robot must evaluate the contextual condition that the source is a care worker. Only if the condition holds, the task can be executed.
- Similarly, as we do not want to state policies tied to a specific task, i.e., a specific sequence of activities, we did not use resulting_conditions to invoke the next activity. However, a policy checker that executes the tasks and checks policies for each activity is needed, as indicated in Figure 5.

Essentially, ACAC itself and our way of using it makes excessive use of conditions: conditions are used to check whether the (smart) environment is in the desired state. Thus, ACAC provides us with a constraint language in which both aspects, robot task planning and access control, can be addressed in a uniform fashion, however, not in a very comfortable one. A potential alternative or addition is to employ a usage control model, as discussed in the next subsection.

3.3 An Attempt based on Usage Control

With usage control, attributes are not only checked before access but also during and after access, and attributes can change as result of accessing a resource. UCON$_{ABC}$ [18] is a family of usage control models integrating authorization, obligations and conditions. We consider the UCON$_{preA_3onA_3preB_3onB_3preC_0onC_0}$ model, that is, including the continuity of decision factors authorization, obligation and conditions and mutability of authorization and obligation before and during access. For simplicity, in the following we refer to this model as UCON.

To model affordances in UCON, they can be viewed as attributes of subjects and objects. Then, for each possible activity, the `allowed` predicate of UCON has to be defined and needs to be evaluated. As with our ACAC examples, the `allowed` predicate has to be evaluated for the set of subjects in contrast to the original UCON model. For instance, if the source asks the robot to bring pain reliever from the supply room, the robot would be physically able to do this task in terms of accessibility (because the supply room has the affordance *moving* and the pain reliever the affordance *grasping*) but if either the source or the robot do not have the right to access medicine, then the task would be denied by access control.

Let us assume again the following subjects and objects:

Subjects: S (Source, i.e., `person`), R (`robot`)

Objects: `bottle`, `table`, `room`, `medicine`, `robot`, `person`, `health_record`

The robot has the right to execute an action if the associated object has a specific affordance, and conditions and obligations are fulfilled. Let O be the set of objects, P the set of persons, and $\mathcal{P}(X)$ the power set of a set X. In our example, objects and subjects have the following attributes:

$$\begin{aligned}
\texttt{alcoholic} &: O \to \{0,1\} \\
\texttt{num_drinks} &: P \to \mathbb{N} \\
\texttt{age} &: P \to \mathbb{N} \\
\texttt{takes_medicine} &: P \to \mathcal{P}(O) \\
\texttt{affordances} &: O \to \mathcal{P}(\{\texttt{grasp, move, receive, ...}\}) \\
\texttt{roles} &: P \to \mathcal{P}(\{\texttt{care_worker, resident, visitor}\})
\end{aligned}$$

For a task $T = [T_1, \ldots T_n]$, a UCON model has to provide the policies to check if the subject is allowed to request the task and if the robot is allowed to execute the task for each activity T_i:

$$\texttt{allowed}((S,R),T) \Rightarrow \texttt{allowed}((S,R),T_1) \wedge \ldots \wedge \texttt{allowed}((S,R),T_n)$$

In the example "Bring me the bottle from my room" (see Section 3.2), the PDP has to query and check the following policies:

$$\begin{aligned}
&\texttt{allowed}((S,R),[\texttt{enter, room}]) \\
&\quad \Rightarrow \texttt{enter} \in \texttt{affordances(room)} \\
&\qquad \wedge \texttt{is_resident(S, room)} \\
&\qquad \wedge \texttt{is_inside(resident(room), room)} \\
&\texttt{allowed}((S,R),[\texttt{move_to, object}]) \\
&\quad \Rightarrow \texttt{move} \in \texttt{affordances(object)} \\
&\texttt{allowed}((S,R),[\texttt{grasp, object}]) \\
&\quad \Rightarrow \texttt{grasp} \in \texttt{affordances(object)} \\
&\texttt{allowed}((S,R),[\texttt{hand_over, object, person}]) \\
&\quad \Rightarrow \texttt{grasp} \in \texttt{affordances(object)} \\
&\qquad \wedge \texttt{receive} \in \texttt{affordances(person)}
\end{aligned}$$

In case the use of an object is particularly restricted, e.g., a bottle because of alcoholic content, a specialized policy where `object=bottle` is used instead of the general policy:

$$\begin{aligned}
&\texttt{allowed}((S,R),[\texttt{hand_over, bottle, person}]) \\
&\quad \Rightarrow \texttt{grasp} \in \texttt{affordances(bottle)} \\
&\qquad \wedge \texttt{take} \in \texttt{affordances(person)} \\
&\qquad \wedge (\texttt{alcoholic(bottle)} \Rightarrow \texttt{age}(S) \geq 21) \\
&\texttt{post_update}(\texttt{num_drinks}(S)) : \\
&\quad \texttt{num_drinks}(S) \leftarrow \texttt{num_drinks}(S) + 1
\end{aligned}$$

Consider a second task where the robot should give information on "Is Bob taking medicine?" which can be translated into a formalized task $T = [[\texttt{obtain, health\_record, bob}]]$ on the virtual object `health_record`:

$$\begin{aligned}
&\texttt{allowed}((S,R),T) \Rightarrow \texttt{allowed}((S,R),[\texttt{obtain, health_record, bob}]) \\
&\texttt{allowed}((S,R),[\texttt{obtain, health_record, person}]) \\
&\quad \Rightarrow \texttt{read} \in \texttt{affordances(health_record)} \\
&\qquad \wedge \texttt{can_access}(S, \texttt{health_record, person})
\end{aligned}$$

The two basic examples illustrate that tasks can be transferred to UCON policies, and then rights can be further refined. However, UCON alone might not be the immediate choice for a JMF. As indicated in [8], usage control might rather be considered "orthogonal" to activity-centric access control. Essentially, using a pure UCON approach, an "allow-listing" of all activities is done in the form of the `allowed` predicate. From a modeling perspective, however, an approach with more structured abstractions might be preferable.

4 RESEARCH AGENDA

In [8], Gupta and Sandhu initiated a discussion and proposed a research agenda on the topic of activity-centric access control for smart connected ecosystems. In our Blue Sky paper, we have followed this "call to action" and contribute first insights into the case of task planning for humanoid robots in assisted living scenarios when considered as activity-centric access control. The case of assistive humanoid robots can help to further develop activity-centric access control by bringing in what has been done for some time in robot task planning, thus shaping the notion of activity in cyber-physical systems. Furthermore, the wealth of interactions in this use case which are outlined as challenges **C1-C4** in Section 1 sets the bar high for architecture & enforcement, modeling, and policies & objectives layers. We structure the research agenda towards the overall goal of safe, secure, and privacy-aware assisted living according to these three layers and emphasize a *joint* treatment of both, robot task planning/execution and access/usage control.

4.1 Joint Modeling Framework

A Joint Modeling Framework should allow modeling from both, robotics and information security and privacy, perspectives in a uniform way to ease the treatment of dependencies between making something feasible and protecting resources. We have shown that the concepts of tasks and affordances used in robot task planning can be mapped to activity-centric access control. However, we have also seen that important aspects might be possible but somewhat tedious to model and various modeling challenges exist:

Cooperative tasks: typical tasks in assisted living involve various subjects and objects. Thus, it would be nice to model tasks and activities directly with several subjects and objects. However, how can invariants or pre-/post-conditions then be easily modeled and still allow formal verification?

Delegated tasks and commands as authorizations: commands to a robot might be ad-hoc authorizations/delegations that dynamically adjust the policy set. Thus, these ad-hoc authorizations need to be checked for conflicts against a priori policies as well as against environmental conditions. Modeling should allow for efficient conflict checks and/or resolution.

Generalized affordances: we generalized the concept of affordances to virtual objects (e.g., healthcare record) and to subjects (e.g., looking affordance, i.e., the robot can perform a "look left" action on itself) and locations (e.g., moving affordance of a room, i.e., the robot can move into a room). Is this a reasonable generalization or an overloaded concept?

Granularity of task description and of interaction between cognitive and access/usage control architecture: on the one hand, separation of concerns represents a reasonable design pattern also for the requested joint modeling framework, on the other hand, due to the abilities of the subjects, there is the danger that subjects are able to fulfill conditions that should not be satisfiable at all. This issue shows some similarity with re-entrancy in the world of smart contracts [16]. What modeling approach would facilitate such re-entrancy analyses?

Learning: robots will learn over time. How is this learning reflected in access and usage control modeling aspects? What kind of reasoning on access or usage rights is the robot allowed to do?

Privacy constraints: privacy aspects need to be included in the modeling framework as well. Privacy, as the term suggests, is definitely context-dependent, thus, cannot be simply separated.

Model overload: taking all the above modeling requirements together, is it reasonable to build one single 'unifying' model? Or is it more about joining models? So far, we translated task planning to access/usage control models. One could also think of the other direction or of a true joint model.

In the end, a task is a series of actions on objects, and actions can be chosen as the "atomic level" of an access. While access control systems like UCON usually base their decisions on this level of access, ACAC already goes forward in the direction of taking the context of an activity, in particular the other activities inside a task, into account to decide whether the task as a whole is allowed. ACAC already contains the concept of activity-based preconditions, but in the end, in order to make policies less circumventable, access control would have to be stated not only on the level of activities, but also on the level of allowed world state after task execution. For example, if a robot is not allowed to hand over a box of pain relievers to certain sources, that policy could be circumvented by commanding the robot to place the box on a nearby table instead and the source taking them by themselves. This could be prevented with a policy that demands that the medication is not allowed to be brought into accessibility of subjects who are denied pain relievers.

4.2 Joint Architecture & Enforcement

A key challenge for architecture and enforcement in the assisted living scenario with one or multiple humanoid robots lies in the tension between sharing data for cooperation and protecting data for privacy and security. In this subsection we focus on data and communication, policies are addressed in the next subsection.

Identifying 'hooks': from a single robot instance's perspective, the underlying problem looks somewhat similar to placing 'hooks' for access control in an operating like it is done in Linux with its Linux security modules. However, the trusted computing base in the addressed scenario spans over multiple robots and other entities.

Access control for a distributed or decentralized Policy Information Point: from the access control system perspective, the robots and other databases form a distributed or decentralized PIP. However, not all information is allowed to flow freely between parts of the PIP, but the information flow is subject to a sharing policies itself. Thus, one has to enforce consistent access control on the (logically centralized) PIP.

A cognitive Policy Information Point: as illustrated in Figure 2, the PIP can potentially make use of all the memories of the humanoid robot and, therefore, could potentially leak most sensitive data. While the use of this most sensitive data might be important for task planning, there is the danger that even when the data is not shared, it can be indirectly observed by actions taken by the robot. Thus, there are open research question in coupling concepts of privacy (like differential privacy concepts) with robot task planning and with their use in the PIP.

Middleware concepts: interaction and cooperation in assisted living require the exchange and possibly the distributed storage of sensitive data. As in other cyber-physical systems, the embedded

world merges with the cloud/edge infrastructure, but now in assisted living the most sensitive data is at stake. For this reason, a decentralized messaging platform like Matrix [17] looks like a promising approach to keep data at the location of assisted living and/or of assistance. Still, access control of decentralized systems like Matrix represents various research opportunities [10], too.

Risk assessment and fault tolerance: we like to add that the sketched architecture needs a high degree of fault tolerance due to their life-affecting aspects. Unlike state machine replication, the various parties and modules do not represent replicas of one single state machine, but come with their specific skills and competencies. This aspect needs to be analyzed for an appropriate risk assessment of proposals for joint architecture and enforcement.

4.3 Joint Policies & Objectives

For any enforcement of policies, the process of creating those policies, i.e., the policy governance, comes first. In the assisted living scenario, we do not have a a single entity that is administering all policies and permissions, but policies are created in consultation among parties. This distributed or even decentralized nature of assisted living leads to various research challenges.

Approval processes and agreement: the necessary agreement of parties with different opinions, needs, powers, and permissions makes a difference in the humanoid assistive robotics case, i.e., there is a need for decentralized policy governance. The policy agreement process is not only run once a priori, but needs to be able to run ad-hoc during runtime whenever agreement is required. For example, a medical doctor, the resident and her/his relatives may come to an ad-hoc agreement on a new medication or diet plan, which the robot (!) has to obey. Relationship-based access control approaches need to be checked for modeling the social connectedness of an assistive humanoid robot and its assisted person.

Software updates: a specific approval process and agreement might be needed to allow updates of the robot's software. As with automotive software, the concept of software identity might be necessary to assign the software with attributes on its behavior.

Policies in the age of learning: having only a restricted set of capabilities, access control on task-specific systems can be done easily by white-listing, i.e., explicitly listing what under which conditions is allowed. In contrast, the task-universality and the learning abilities of humanoid robots imply a need to comply with complex societal rules and regulations. In other words, the robot's capabilities might be too complex to allow for white-listing specific capabilities. Instead, policy creation has to happen dynamically to maintain the robot's task-universality.

Emergency scenario: an emergency situation can be considered as extreme cases of interaction where the robot needs to act on its own. It has to detect emergencies and, then, has to potentially override the privileges of requesting subjects. As a robot should not let people come to harm by inactivity, the robot needs to give elevated priorities and access rights without an explicit command, and might receive elevated priorities and access rights in both helping and calling for help.

When an environment becomes "smart," various previous "objects" now become "subjects." Therefore, aspects of cooperation and agreement become even more important. Reaching agreement is a problem well-known to be hard in the presence of faults, and consistency and availability might be hard to achieve in the presence of network partitions. As some policy decisions require the policy takers to have the same set of information or even consensus between parties, the challenges of joint policies & objectives also represent requirements on the other aspects, architecture & enforcement as well as modeling. We also like to emphasize that various ethical aspects need to be considered when policies and conflict resolution mechanisms are defined for the case of assistive humanoid robots.

4.4 Achievement Levels

Future proposals will be judged by the level of autonomy and task universality achieved together with an associated security and privacy assessment. Like in autonomous driving, levels of autonomous assistance need to be defined as well as levels of assurance of the implemented task and access control.

5 RELATED WORK

We focus on related work on security and access control aspects in the context of robotic systems, smart homes, and automotive systems that we believe is helpful to consult when addressing the proposed research agenda. Coverage of general task planning of humanoid robots is out of scope due to space constraints.

Security Analysis of Robotic Systems. ROS (Robot Operating System) is a popular robotic operating system with an open source app store. Attackers can compromise the security and privacy by uploading malicious software [31] and potentially the safety of humans [6] due to unauthorized access to robotic functions like arm movement. If deploying access control to robot task planning by design as well as software identity management (see Research Agenda), some of those threats can be prevented.

Access Control for Robotics. Access control models for ROS and ROS2 have been proposed [29, 32]. In [32] a policy-based access control (PBAC) to ROS is deployed for security guarantees like data privacy as well as safety guarantees to protect human lives. Since ROS does not provide access control by default, the authors propose to implement PBAC on top of ROS. Using user-determined policies, for every application the permissions to the resources can be managed, similar to Android's approach to restrict access on resources. Their access control model also includes runtime permission revocation. While PBAC represents an important building block, we emphasize the need to be able to obtain permissions by delegation which are revoked after the task has finished. Furthermore, every robot itself should host a policy decision point and a policy information point to eliminate a central access manager.

In [29], the authors propose a framework for systematic generation and verification of cryptographic artefacts. Here the robotic system is part of an IoT system with many nodes where the robotic systems, that is, ROS, is a middleware in the IoT system and the systems form a communication graph. Their framework allows for enforcing access control policies on invoking objects in the communication graph.

Access Control for Smart Home and for Automotive Systems. While smart home systems as well as automotive systems are typically not considered to be task-universal compared to a humanoid robot,

various aspects overlap between these domains. In [23] a system for multi-user multi-device-aware access control for smart homes is proposed that particularly focuses on policy negotiation and conflict resolution. The negotiation and conflict resolution mechanisms might also serve the assistive humanoid robot scenario, however, as outlined in the Research Agenda, a more decentralized approach might be preferable. The need to provide a clearly defined access control interface between embedded components and their connection to external services for automotive systems is addressed, for example, in [13] and in [20]. These proposals might help for the definition of the interface between a robot's memories and the policy information point.

6 CONCLUSION

In this Blue Sky paper, we explored the relationship between task planning and access control when applied to assisted living with humanoid robots. We identified task universality, public and private operations areas, and the social connectedness of these assistive humanoid robots as the main challenges for safe, secure, and privacy-aware assistance. We indicated how the cognitive architecture of such a robot interacts with an access and usage control architecture, and explored how the notions of affordances and activities can be modeled in activity-centric access control as well as in a usage control model. Clearly, activity-centric access control provides a promising approach to integrate robot task planning with access control. Based on first insights, we proposed a research agenda for the three perspectives addressed in this paper: architecture & enforcement, modeling, and policy & objectives. These three perspectives are related to well-known 'grand' challenges: bringing together *i)* the worlds of embedded and cloud/edge computing, *ii)* allowing a joint treatment of safety, security, and privacy-awareness for assistive services, and *iii)* agreeing on policies and objectives in a socially connected online world. While these challenges are shared by various application domains, the case of assistive humanoid robots looks to be specifically demanding and, therefore, paradigmatic for future research.

Acknowledgements. This work has been supported by the project "Stay young with robots" (JuBot). The JuBot project was made possible by funding from the Carl Zeiss Foundation.

REFERENCES

[1] Paola Ardón, Èric Pairet, Katrin S. Lohan, Subramanian Ramamoorthy, and Ronald P. A. Petrick. 2020. Affordances in Robotic Tasks – A Survey. arXiv:2004.07400 [cs.RO]

[2] Tamim Asfour et al. 2019. ARMAR-6: A high-performance humanoid for human-robot collaboration in real-world scenarios. *IEEE Robotics & Automation Magazine* 26, 4 (2019), 108–121.

[3] Tamim Asfour, Rüdiger Dillmann, Nikolaus Vahrenkamp, Martin Do, Mirko Wächter, Christian Mandery, Peter Kaiser, Manfred Kröhnert, and Markus Grotz. 2017. The Karlsruhe ARMAR Humanoid Robot Family. In: Goswami A., Vadakkepat P. (eds) Humanoid Robotics: A Reference, Springer, 1–32.

[4] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. 2008. Robot task planning using semantic maps. *Robotics and Autonomous Systems* 56, 11 (2008), 955–966. https://doi.org/10.1016/j.robot.2008.08.007 Semantic Knowledge in Robotics.

[5] Sergio Garcıa, Claudio Menghi, Patrizio Pelliccione, Thorsten Berger, and Rebekka Wohlrab. 2018. An Architecture for Decentralized, Collaborative, and Autonomous Robots. In *2018 IEEE International Conference on Software Architecture (ICSA)*. 75–7509. https://doi.org/10.1109/ICSA.2018.00017

[6] A. Giaretta, M. De Donno, and N. Dragoni. 2018. Adding salt to pepper: A structured security assessment over a humanoid robot. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. 1–8.

[7] James J. Gibson. [n. d.]. The theory of affordances. In *Perceiving, Acting, and Knowing: Toward an ecological psychology*. Lawrence Erlbaum Associates, 67–82.

[8] Maanak Gupta and Ravi Sandhu. 2021. Towards Activity-Centric Access Control for Smart Collaborative Ecosystems. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*. 155–164.

[9] Marc Hanheide, Moritz Göbelbecker, Graham S Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, et al. 2017. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence* 247 (2017), 119–150.

[10] Florian Jacob et al. 2020. Matrix decomposition: analysis of an access control approach on transaction-based DAGs without finality. In *Proc. of the 25th ACM Symposium on Access Control Models and Technologies* (Barcelona, Spain) *(SACMAT '20)*. ACM, New York, NY, USA, 81–92. https://doi.org/10.1145/3381991.3395399

[11] Yuqian Jiang et al. 2018. LAAIR: A Layered Architecture for Autonomous Interactive Robots. *ArXiv* abs/1811.03563 (2018).

[12] Yuqian Jiang, Shiqi Zhang, Piyush Khandelwal, and Peter Stone. 2019. Task Planning in Robotics: an Empirical Comparison of PDDL-based and ASP-based Systems. arXiv:1804.08229 [cs.AI]

[13] Dae-Kyoo Kim, Eunjee Song, and Huafeng Yu. 2016. *Introducing attribute-based access control to AUTOSAR*. Technical Report. SAE Technical Paper.

[14] Hadas Kress-Gazit et al. 2021. Formalizing and Guaranteeing Human-Robot Interaction. *Commun. ACM* 64, 9 (2021), 78–84. https://doi.org/10.1145/3433637

[15] Norbert Krüger, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrčen, Alejandro Agostini, and Rüdiger Dillmann. 2011. Object–Action Complexes: Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems* 59, 10 (2011), 740–757. https://doi.org/10.1016/j.robot.2011.05.009

[16] Chao Liu, Han Liu, Zhao Cao, Zhong Chen, Bangdao Chen, and Bill Roscoe. 2018. ReGuard: Finding Reentrancy Bugs in Smart Contracts. In *2018 IEEE/ACM 40th Int. Conf. on Software Engineering: Companion (ICSE-Companion)*. 65–68.

[17] Matrix.org Foundation C.I.C. 2021. *Matrix specification v1.1*. Technical Report. https://spec.matrix.org/v1.1/.

[18] Jaehong Park and Ravi Sandhu. 2002. The ABC Core Model for Usage Control: Integrating Authorizations, oBligations, and Conditions. *ACM Transactions on Information and System Security* 2, 3 (2002), 09.

[19] Laurel D. Riek. 2017. Healthcare Robotics. *Commun. ACM* 60, 11 (oct 2017), 68–78. https://doi.org/10.1145/3127874

[20] Marcel Rumez, Alexander Duda, Patrick Gründer, Reiner Kriesten, and Eric Sax. 2019. Integration of attribute-based access control into automotive architectures. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1916–1922.

[21] Ravi Sandhu. 2000. Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way. In *Proc. ACM Workshop on Role-Based Access Control* (Berlin, Germany). ACM, New York, NY, USA, 111–119. https://doi.org/10.1145/344287.344309

[22] Ravi Sandhu. 2009. The PEI framework for application-centric security. In *2009 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing*. 1–5. https://doi.org/10.4108/ICST.COLLABORATECOM2009.8382

[23] Amit Kumar Sikder, Leonardo Babun, Z Berkay Celik, Abbas Acar, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A Selcuk Uluagac. 2020. Kratos: Multi-user multi-device-aware access control system for the smart home. In *Proc. of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 1–12.

[24] Anna Cinzia Squicciarini. 2020. Multi-party Access Control-10 Years of Successes and Lessons Learned. In *Proceedings of the 25th ACM Symposium on Access Control Models and Technologies*. 189–190.

[25] Nikolaus Vahrenkamp, Mirko Wächter, Manfred Kröhnert, Kai Welke, and Tamim Asfour. 2015. The robot software framework ArmarX. *it - Information Technology* 57 (01 2015). https://doi.org/10.1515/itit-2014-1066

[26] David Vernon. 2014. *Artificial Cognitive Systems: A Primer*. The MIT Press.

[27] Mirko Wächter et al. 2018. Integrating multi-purpose natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots. *Robotics and autonomous systems* 99 (2018), 148–165.

[28] Mirko Wächter, Simon Ottenhaus, Manfred Kröhnert, Nikolaus Vahrenkamp, and Tamim Asfour. 2016. The ArmarX Statechart Concept: Graphical Programming of Robot Behaviour. *Frontiers in Robotics and AI* 3 (2016), 0–0.

[29] Ruffin White, Henrik I Christensen, Gianluca Caiazza, and Agostino Cortesi. 2018. Procedurally provisioned access control for robotic systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1–9.

[30] Jan Winkler, Georg Bartels, Lorenz Mösenlechner, and Michael Beetz. 2012. Knowledge enabled high-level task abstraction and execution. In *First Annual Conference on Advances in Cognitive Systems*, Vol. 2. Citeseer, 131–148.

[31] Yuan Xu, Tianwei Zhang, and Yungang Bao. 2021. Analysis and Mitigation of Function Interaction Risks in Robot Apps. In *24th International Symposium on Research in Attacks, Intrusions and Defenses*. 1–16.

[32] Yi Zong, Yao Guo, and Xiangqun Chen. 2019. Policy-based access control for robotic applications. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 368–3685.