Formal Ontology in Information Systems
N. Aussenac-Gilles et al. (Eds.)

doi:10.3233/FAIA231137

Towards an Ontology for Robot Introspection and Metacognition

Robin NOLTE [a], Mihai POMARLAN [b], Daniel BESSLER [c], Robert PORZEL [a], Rainer MALAKA [a], and John A. BATEMAN [b]

[a] *University of Bremen, Digital Media Lab, Bremen, Germany*
[b] *University of Bremen, Department of Linguistics, Bremen, Germany*
[c] *University of Bremen, Institute for Artificial Intelligence, Bremen, Germany*

Abstract. We present the *Meta-Ontology for Introspection (MOI)*: Inspired by fundamental processes of the human mind, cognitive architectures (CAs) explore ever more methods to leverage metacognition. Still, an ontological model to trace metacognitive experiences for learning or as input for metacognitive control routines has yet to be developed. Based on a review of existing standards, we formally identify the relevant scope in the form of Competency Questions (CQs) and extend SOMA, a well-established formal ontology initially designed to interpret episodic memories of a robotic CA. The resulting MOI can model a CA's software and capabilities of single components, trace information processing and inter-component communication, label self-lived mental events, and capture causal relationships. We evaluate MOI via the CQs and exemplarily demonstrate its reasoning capabilities.

Keywords. Domain Ontology, Introspection, Metacognitive Experiences, Tracing

1. Introduction

Metacognition is fundamental for human cognition, covering "any knowledge or cognitive process that is involved in the interpretation, monitoring or control of cognition" [1]. This entails two levels of cognition: The *meta-level* controls the basic *object-level* using data gathered from constant monitoring [2]. This includes *metacognitive knowledge*, which holds facts about one's own cognition (e.g., "I know how to solve for x."), *metacognitive control strategies*, which are subconscious behaviors to manipulate one's own mind (e.g., the habit of inventing mnemonics), and *metacognitive experiences*, interpretations of one's own mental state (e.g., experiencing and labeling a déjà vu) [3].

A growing number of computational *cognitive architectures (CAs)* also employ principles inspired by human metacognition [4]. A common technique is to trace employed (reasoning) algorithms for online evaluation or offline learning. We consider a scenario where different reasoners cooperate in answering queries hybridly, e.g., to serve a robotic agent with the necessary information to solve complex tasks. Each system could be tailored to different accuracy, answer speed, or domains of expertise. To evaluate the reliability of conclusions or to trace back an error's source, it is necessary to track their information flow over time, annotated with metadata such as correctness of results. Metaphorically, the CA's metacognitive experiences have to be modeled and stored.

Although self-lived experiences are usually associated with *episodic memory*, no consensus has been achieved on episodic memory for CAs [5]. Instead, episodic memory "has been largely ignored by CAs" [6] and "remains relatively neglected in computational models of cognition" [4]. One exception to this [7] is the *Cognitive Robot Abstraction Machine (CRAM)* [8] and its reasoning engine *KnowRob* [9], which embrace knowledge structures known as *narrative-enabled episodic memories (NEEMs)* [9]. NEEMs link sub-symbolic data about the robots' physical experiences to the robotic agent's interpretation thereof, which is expressed in a formal ontological model. For example, quantitative data on pose transitions relate to motion events that might be classified by the task of cutting bread. To our knowledge, this feature is unique to CRAM and KnowRob. As capturing single-purpose training data with robots is time-consuming, NEEMs are constantly logged. For different learning tasks, selected parts can then later be queried via the free *Open-EASE* platform [10]. This has proved useful, e.g., for learning action parameterization [11,12], learning common-sense knowledge from humans in VR [13], and transferring experiences between robots and affordances to novel objects [14].

While there are both examples of CAs employing formal ontologies, especially in the robot domain (see [15,16]), and of CAs that use tracing for metacognition (see [4]), we are not aware of any ontological model to express such traces. Although NEEMs have been used to log messages between the CRAM executive and KnowRob in a recent experiment on learning how to self-specialize plans [17], this approach towards ontology-enabled introspection was unsystematic and only covered specific reasoning events. Others have logged perception events using *ARBI* and *ISRO* [18]. However, these approaches do not come close to cognition tracing: Instead, the complete message flow between all cognitive components has to be tracked – potentially extending even to details of their internal information processing. This lack of a corresponding domain ontology is surprising, especially given the robot community's ongoing efforts to make research data openly accessible, e.g., by standardizing terminology via formal ontologies [19,20,21], or by hosting them on research platforms such as Open-EASE or RoboEarth [22].

Here we propose the formal *Meta-Ontology for Introspection (MOI)* to model and trace metacognitive experiences, extending the *Socio-Physical Model of Activities (SOMA)* ontology [21,23], which itself builds on the upper-level ontology *DOLCE+DnS Ultralite (DUL)* [24,25]. We show that this extension is natural since SOMA is explicitly designed to capture episodic memory and has been employed for this with great success, e.g., as the underlying axiomatization of NEEMs. Note that we do not claim MOI to accurately represent *human* cognition, as it instead concerns computational processes; such research would require a detailed analysis of proposals for human CAs as well.

Our methodology roughly follows the SABiO guidelines for ontology development [26]: We first define the scope via *competency questions (CQs)* [27] in Section 2. In Section 3, we give an overview of relevant existing models, covering introspection, CA experiences, capabilities and software. Section 4 briefly introduces SOMA and then presents its novel extension MOI. In Section 5, we evaluate our model via the defined CQs.

2. Scope

As per the SABiO guidelines [26], ontology engineers, potential users and domain experts collaboratively defined MOI's scope. Due to their unique experience in working

with ontology-based episodic memories, we selected CRAM, KnowRob, and OpenEASE developers to constitute both the pertinent domain experts and the future users. The main purpose of tracing a CA's metacognitive experiences naturally requires the ontology to model the telemetry between all software components. However, component-specific information is relevant as well: What data do they hold and process, or, more abstractly, what mental events do they experience? For this, we need taxonomies of communication and mental tasks, together with associated roles, e.g., to label sender and receiver of a message, or information as premises and conclusions.

Inspired by software tracing tools, we aim for a flexible level of abstraction: It should be possible to trace abstract communication and mental events of cognitive components like memory and perception, and of software components like server and clients, as well as atomic method calls and algorithm steps. For this, our model must both compose complex processes from sub-processes and include basic patterns to represent software. Besides event composition, tracing tools commonly relate events via causality. These principles offer rich semantics that typical logging frameworks lack. To see the benefits, consider highly parallelized systems with multiple agents communicating simultaneously. As multiple messages could have similar time stamps, a server's log about receiving a query is hard to associate with a client's log about sending that query if both were only to contain uninterpreted text. Ergo, a basic model of causality is in scope as well.

A secondary goal is to provide some metacognitive knowledge in the form of (mental) *capabilities*, as control routines might require information on a cognitive routine's abilities to perform specific actions and its qualities in doing so. For example, as explained in the introduction, control functions need to know what object-level routines can reason about what domain and how reliable their inferences are.

Following SABiO, we concretized the scope via narrowed-down *competency questions (CQs)* [27,26] that our model should answer, which then drove the development process iteratively.

CQ1: How do events relate by cause and composition?

CQ2: Via what tasks do agents communicate, and what roles do the participants play?

CQ3: What information does an agent have at what time?

CQ4: Via what tasks do agents process information ("think"), and what roles do the participants play?

CQ5: How does information flow and transform within a CA?

CQ6: How does "thought" relate to the "physical" actions of an agent?

CQ7: What is software, both during and outside of runtime?

CQ8: How does software relate to employed algorithms and the executing machine(s)?

CQ9: What are the capabilities of an agent, and, in particular, a software controller?

We limit the CQs' scope to the domain of computational CAs. For example, we refrain from modeling every speech act of human communication and instead only consider those necessary to model the telemetry between a CA's software components. We nevertheless aim for extendability, e.g., for when human-robot cooperation enters the loop.

3. Related Work

3.1. Ontologies for Introspection

Inspired by the success of bio-medical ontologies, there has been an increasing interest in developing formal ontologies for psychology [28]. While the *Cognitive Atlas* [29] as an extensive knowledge base of cognitive neuroscience is probably the most famous approach, its organization remains shallow. For example, it defines decision-making, facial expression, and hedonism all as siblings. Thus, reasoning tasks relevant to our case are not supported. Others model specific sub-domains of psychology, e.g., mental illness, EEG, and emotion [30]. Most recently, *IM-Onto* was developed to homogenize the vocabulary of metacognition across different research fields [31]. As it focuses on meta-reasoning tasks and only considers a single object-level task (planning), it is of little use for labeling object-level cognition.

Metzler and Shea (2011) surveyed which capabilities researchers in the field of CAs deem relevant for CAs [18] and developed an informal taxonomy of "mental capabilities." Although they intended to assist CA development, they also exemplarily classified the cognitive steps involved in a robot's route planning. The capabilities considered are somewhat abstract, e.g., to "know," "perceive," and "plan," and the taxonomy lacks more concrete mental functions such as physics simulation as a means for planning.

Similarly to how CRAM and KnowRob exploit SOMA, the *Artificial robot brain intelligence (ARBI)* employs the *Intelligent Service Robot Ontology (ISRO)* [32]. ISRO contains a detailed taxonomy of perception events of a service robot – most other mental events, such as planning, learning, or forgetting, are not considered. As mentioned in Section 1, ARBI logs perception events using ISRO, although telemetry, ensuing data processing and consequences are not logged. Ergo, this can only be viewed as an immature first step towards tracing mental events using ontologies.

Other ontologies concerning aspects of metacognition are not helpful to us as their domains do not cover object-level cognition tasks, but, e.g., metacognitive strategies to automatically replace failing components [33,34] or possible responses of meta-level functions to certain diagnoses [35,36]. In summary, only the taxonomy by Metzler and Shea structures general concepts of object-level cognitive tasks, although ISRO and IM-Onto can be referred to for perception and metacognitive events, respectively.

3.2. Ontological Models of CA Experiences

We focus here solely on ontologies concerning the experiences of robotic CAs. Olivares-Alarcos et al. (2019) [15] and Manzoor et al. (2021) [16] survey robotic frameworks employing formal ontologies. None of these support tracing or the collection of telemetry data, although ISRO logs perception events (see Section 3.1). Furthermore, KnowRob with SOMA is the only framework that aims to fully support episodic memory. SOMA describes physical actions in detail but, thus far, lacks mental tasks and communication.

The AWARE ontology [37] describes observations and decisions of a mobile robot, but as well fails to explain how they relate. The *Deontic Cognitive Event Ontology (DCEO)* models the mental states of agents for interaction, e.g., perception, beliefs, desires and intentions [38]. DCEO does not support our use case, however, as it associates those mental states with the owner in a highly abstract way, neglecting all internal control systems, which makes it impossible to trace the information flow.

3.3. Capabilities

Capabilities are closely related to *Dispositions*, which portray an object's proneness to participate in particular processes [23]. For example, a knife may have the disposition to cut bread, but not trees. Merrell et al. (2019) define capabilities as benefit-bearing dispositions and sort them between the *Basic Formal Ontology (BFO)* [39] classes of `Disposition` and `Function` [40]. In our opinion, this falls short insofar that agents can also harm themselves and must continually avoid doing so, e.g., by exercising caution.

Capabilities (also skills and functions) are common in robot ontologies, e.g., in ISRO and TOMASys [34]; for others, see recent surveys [15,16]. ISRO associates capabilities with robots only, and cannot express, e.g., that a database has the capability to answer queries. Although not covering capabilities, SOMA includes an elaborate dispositional model of *affordances* [23] that describes how dispositions interact. For example, for a cutting affordance to manifest, the dispositions of something (suitably) sharp and something (suitably) cuttable must combine. This is encoded by the relations `affordsBearer` and `affordsTrigger` describing what `Roles` the bearer, i.e., the dispositions' owner, and the triggers, i.e., objects with matching dispositions, play in the associated task. In the cutting example, the disposition of a knife affords the bearer to cut and the trigger to be cuttable. Another disposition matches this if the roles of bearer and trigger are switched, and, additionally, both dispositions afford the same task.

3.4. Software Models

While the problem of describing software is relevant for robotics, robotics ontologies (see Section 3.1) have so far not focused much on this and most consider only single concepts such as algorithms or software. Exceptionally, ISRO and SOMA's predecessor, the *Semantic Robot Description Language (SRDL)* [41], taxonomize software components but without associated tasks or a model of what software is or how it relates to algorithms. A conceptual model of software is described in [42], which differentiates between code, programs, software systems, and software products, where each preceding constitutes the succeeding; this work is however not axiomatized.

Several ontologies specifically target software, such as the *Core Software Ontology (CSO)* [43], *Core Ontology of Programs and Software (COPS)* [44] (both of these use DOLCE [24] as a foundation), and the *Software Ontology (SwO)* [45]. They make distinctions between entities such as algorithms, source code, source code files and running software instances. Importantly, these models treat running software instances as perdurants, e.g., as computational activity, (CSO, COPS) or as the computer's dispositions (SwO). Similarly, de Oliveira views software capabilities as borne by the executing computer [46], not the running instances.

While the above conceptualizations of software instances are valid, they remain insufficient in our case as features of running software that abstract away from its physical manifestation are not captured. We must describe properties associated with agents, e.g., the ability to execute tasks, and to associate running software with the capabilities to communicate and process information. Further, in our use-case, a running piece of software must be able to participate in events, but not be an event itself. As such, our model, especially of running software, must be of a medium level of abstraction that is not yet found in existing software ontologies, sitting between the low-level ontologies

CSO and COPS, and the high-level models of SwO and de Oliveira. To our knowledge, the only considerations of the agentive features of running software so far are a comment contained in the implementation of DUL stating that "a computational agent can be considered as a PhysicalAgent," and the `SoftwareAgent` concept in the upper ontology YAMATO [47].

4. Model

4.1. Ontological Grounding

We now present the *Meta-Ontology for Introspection (MOI)* as a model to trace metacognitive experiences of (robotic) CAs. We decided to extend the formal *Web Ontology Language (OWL)* [48] ontology SOMA [21], which is well established and explicitly designed to express and log robot experiences via many patterns.

For example, in the robotic domain, particularly in bridging physical events to concepts thereof, it is imperative to recognise that *interpretations* take place: Any characterization of an objective occurrence unexceptionally depends on the observers' subjective narrative [49]. Such an interpretive view has been employed with great success in SOMA-flavoured NEEMs [10,11,12,13,14] and has also been argued to be propitious for classifying mental processes [50]. SOMA enforces this stance by building upon the foundational ontology DUL, which consistently distinguishes `PhysicalEntities` from `SocialEntities` that exist "for the sake of [...] contextualizing or interpreting existing entities" [25] such as `Description` and `Concept`. Accordingly, a physical `Event` (an objective occurrence) `executes` an `EventType` (a subclass of `Concept` and thereby a subjective interpretation of said event), that `isDefinedIn` some `Description` such as an observer's `Narrative` or an emerging `Affordance`. More concretely, a `CommunicationAction` may `execute` a `Querying` or `Answering` task as depicted in Figure 1, and a movement may `execute` a cutting `Task`.

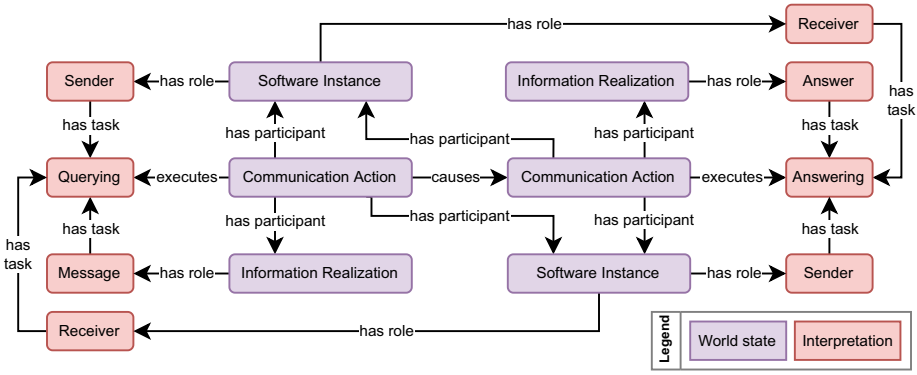

Figure 1. ABox of a `Querying` task that `causes` an `Answering` task. Individuals are labeled with extending concepts and color-coded by association with objective phenomena or "interpretations" thereof.

DUL applies the above principle to other common ontology patterns such as `Roles`, which it also views as a subclass of `Concept` and therefore as defined by some `De-`

scription. Any object *x* that – objectively – isParticipantIn some Event that executes a Task *y* may be associated with a subjective Role linking to *x* via hasRole and to *y* per hasTask as is standard. This allows for contextualizing entities (see, e.g., [51] for details), for example, to differentiate a cutting tool from the object being cut, and a message's sender from its receiver (see Figure 1). Furthermore, however, linking Roles to Narratives can capture that agents disagree on who plays which role, e.g., when detectives argue about the perpetrator's identity. Figure 2 depicts the general pattern.

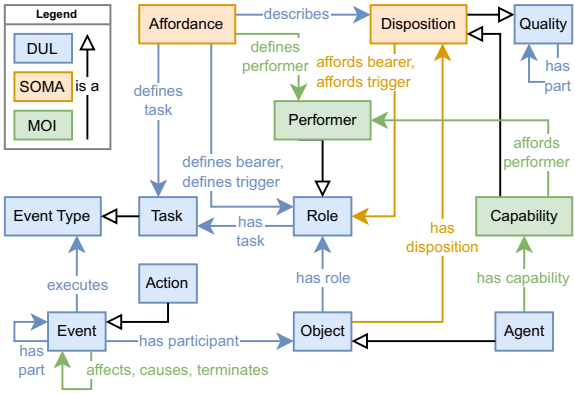

Figure 2. Basic concepts and their relations of the proposed capability model (TBox); color-coded by defining ontology.

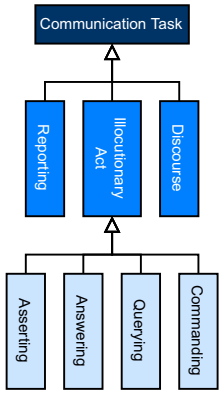

Figure 3. Proposed taxonomy of communication tasks (TBox).

Besides the patterns inherited from DUL, SOMA comes with a dispositional model of affordances as described in Section 3.3, an elaborate taxonomy of everyday activities, and means to describe plans thereof. The latter proves useful as it allows, e.g., to associate mental planning tasks with the subsequently executed actions.

MOI is bundled in recent SOMA versions and distributed over its sub-ontologies following SOMA's modular structure (see [21]). For example, SOMA-ACT contains the new taxonomies of mental- and communication tasks. SOMA, including MOI, is published under the GNU Lesser General Public License Version 3.0 (GPLv3)[1]. Note that although their developers helped to formulate the scope, SOMA and MOI as general schemas are completely agnostic towards CRAM, KnowRob and Open-EASE.

4.2. Telemetry

CQ1 requires a model of causality and composition between events of arbitrary abstraction. The latter can be represented out-of-the-box via DUL's hasPart relation as depicted in Figure 2, e.g., a complex action might have atomic motion and thinking as parts. Regarding causality, the temporal relations offered by SOMA such as before and after are insufficient (see Section 2). To solve this, we say that an Event directlyCauses (isTerminatedBy) another if the latter would not have occurred (ended) in the absence of the former. We additionally introduce a more general relationship: an Event affects another if a variation in the course or outcome of the former would have resulted in a

[1] SOMA, including MOI, is freely available at: https://github.com/ease-crc/soma

variation in the latter. For example, a planning task that specifies a target position affects the subsequent pick-and-place task that uses this parameter.

We argue that causality is transitive, while termination is not. A knocked over domino causes the fall of each subsequent domino in a chain reaction, but in contrast, extinguishing a fire does not transitively cancel the concert that the outbreak interrupted hours ago. Therefore, only for `directlyCauses` do we introduce a transitive superproperty `causes`, which we in turn sort beneath SOMA's `before` due to the intuition that a cause has to occur sooner than its effect. Figure 1 above depicts an exemplary causal relationship between a querying and answering action; Figure 2 shows how the relations embed into DUL's `Action` model.

For CQ2, we add a simple taxonomy of abstract communication tasks (Figure 3) that is straightforward to extend. These cover illocutionary acts [52] in which an agent orders another to execute some instructions (`Commanding`), demands information from another (`Querying`) or responds (`Answering`). Inspired by simple communication models, e.g., Shannon-Weaver [53], we associate the tasks with specific roles. The participants of an `Action` classified as an `IllocutionaryAct` play the roles of `Sender`, `Receiver`, and `Message`, respectively. Furthermore, we model two non-atomic communication tasks that are constituted of others: `Discourse`, in which the participants alternate between the roles of sender and receiver, and `Reporting`, with stable roles.

To answer CQ3, we need to model the information that an agent has. DUL differentiates between `InformationObject`, e.g., the text of a shopping list, `InformationRealizations`, which are embodied copies of `InformationOject`, e.g., a piece of paper with the shopping list written down, and the `SocialObject` expressed by the `InformationObject`, e.g., the sorted collection of items to buy. This pattern also shapes the models of CSO and COPS (see Section 3.4) as DUL is a restriction of DOLCE. CQ3 mainly addresses `InformationRealization`, as these are physically located (within an agent's memory). This can be exploited by SOMA's containment pattern, originally developed for physical containers like kitchen cabinets. An agent then knows a piece of information if it is contained within its memory. In SOMA, this is represented by a `ContainmentState` with the associated roles of a `Container` and a `ContainedObject`.

This allows the system to track changes in an agent's information using `StateTransitions`, which fills the missing piece for answering CQ5. Our only additions are to make the relation between the initial and the terminal state of a transition explicit by introducing the property `replacedBy` and to add super-property chain axioms to automatically infer the new relations `replacedBy`, `terminates`, and `causes` between the `MentalAction` that causes the transition and the initial and terminal `PhysicalStates`.

An exemplary ABox of this pattern is depicted in Figure 4. The mental action of `Memorizing` triggers a `StateTransition` between the "not-knowing" `ContainmentState`, in which the memorized `InformationRealization` is excluded from the `Memory`, to the "knowing" state where it is inserted. For simplicity, Figure 4 does not depict the memorizing agent, the relation between the action participants (agent, memory, `InformationRealization`) or the ownership relation between agent and memory. The `InformationRealization` may also have further relations to `InformationObjects` and `SocialObjects` that represent its content. Information for which OWL is not designed to express, such as pictures and sound, can be referred to via database pointers.

Finally, this approach must not be taken as advocating a monistic view of the mind-body problem in human cognition; recall, instead, that we only consider computational

CAs, for which the adopted metaphor is appropriate because "cognition" manifests itself physically, for example, as files in data storage. Our approach allows for sufficient abstraction even when dealing with exceptions such as black-box models, e.g., neural networks, in which realizations of knowledge are untraceable: those can simply play as a whole the role of the memory in a containment state.

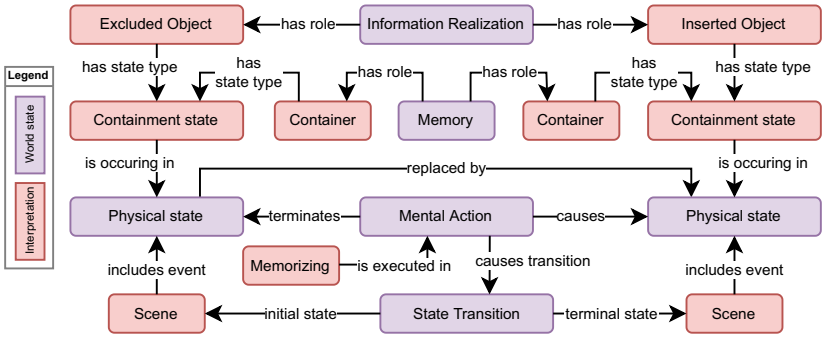

Figure 4. ABox of a `StateTransition` between two `ContainmentStates` (simplified). Individuals are labeled with extending concepts and color-coded by association with objective phenomena or "interpretation".

The causality relations introduced so far and the communication taxonomy capture *inter-agent* communication telemetry. To model *internal* information processing in answer to CQ4, i.e., 'thought', we added an abstract taxonomy of mental tasks as depicted by Figure 5. The taxonomy draws from Metzler and Shea [18] and thereby covers essential mental events from the cognitive literature. We introduced additional upper-level categories based on the roles that participating `InformationRealizations` play. `InformationAcquisition` covers events that output some `Knowledge`, e.g., via `InformationRetrieval` and `DerivingInformation`. The former describes recalling information from memory (e.g., via `Remembering`), while the latter strictly derives `Conclusions` from `Premises` (e.g., via `Reasoning` or `Interpreting`). While `InformationStorage` includes processes that save some information with the roles `StoredObject` and `Knowledge` for later use, e.g., `Learning` or `Memorizing`, `InformationDismissal` in contrast removes some `ExcludedObject` from memory, e.g., `Forgetting`. We also add further detail to model operations that are common in CAs. For example, `Retro-` and `Prospecting` denote attempts to construct representations of past or future states.

Note that for CQ3, `Interpreting`, by which we mean the task of sense making, e.g., compiling a written text into a mental representation of its meaning, raises the question of whether a difference between having information available and understanding it should be modeled. However, in the domain of CAs, we argue that it is so far sufficient to represent a "mental representation of its meaning" as just another `InformationObject`.

For CQ5, we add the property `directlyDerivedFrom` and its transitive variant `derivedFrom` to model that some `InformationObject` was developed from others. We can achieve an automatic deduction of `directlyDerivedFrom` via a complex chain of properties as depicted in Figure 6, where `isInputRoleOf` and `isOutputRoleOf` are sub-properties of `hasTask` and `isTaskOf`, respectively, denoting task input and output.

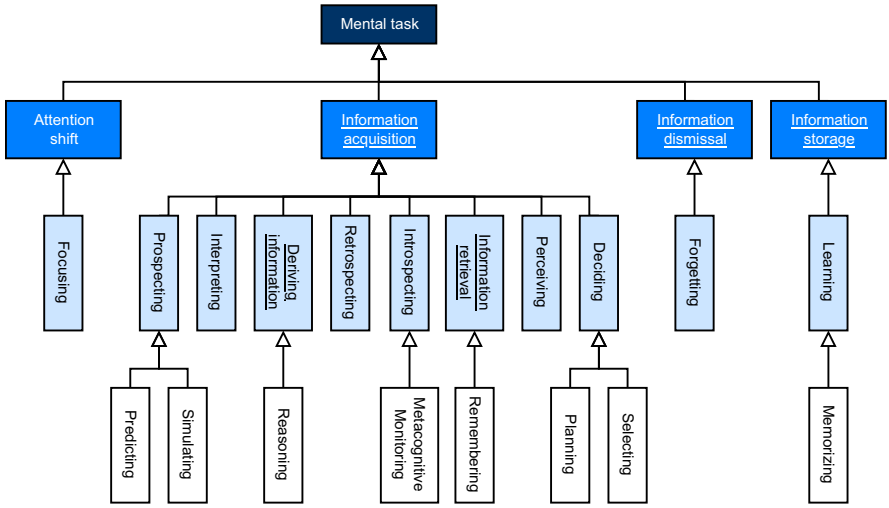

Figure 5. Proposed taxonomy of mental tasks (TBox). Marked concepts are axiomatized w.r.t. relevant roles.

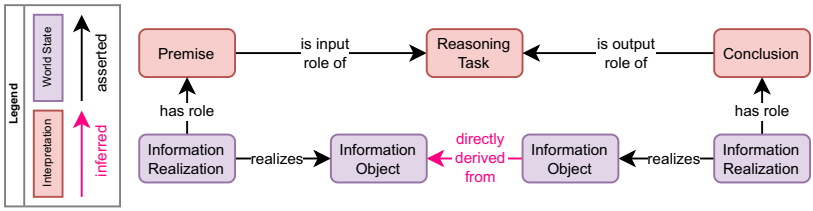

Figure 6. An exemplary ABox of as `ReasoningTask` showcasing the entailment of `directlyDerivedFrom` between two `InformationObjects` via a complex property chain. Individuals are labeled with extending concepts and color-coded by association with objective phenomena or "interpretations" thereof.

4.3. Action execution

Answering CQ6 proved more complex, involving what some deem "to be the fundamental question of philosophy of action" [54]: What is the mental process that brings about a physical action? Although there is evidence mental states impact movement, it is unknown how [55]. In strict *Enactivism*, the question is rejected altogether by the central argument that "cognition comes from bodily action and serves bodily action, that is, cognition is embodied action" [56].

As a pragmatic solution, we do not introduce some execution task, but instead model the relation between a `Planning` task and the planned tasks' execution. A `Planning` task constructs an `InformationObject` that `expresses` some `Workflow`, i.e., a structured `Plan`. The `Workflow` `defines` `Tasks` and `Roles`, which can `classify` (later) actions and their participants. Performing the defined `Tasks` represents the execution of the plan.

4.4. Software Agents

For CQ7 and CQ8, we model software components and their embodiment during runtime based on our survey from Section 3.4; the result is depicted in Figure 7. Recall that

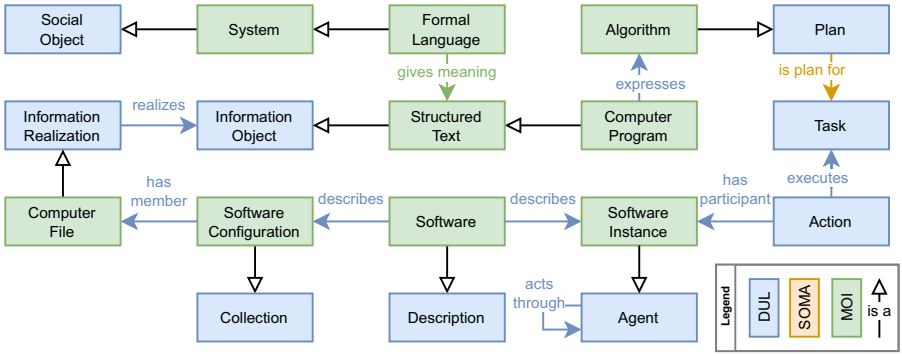

Figure 7. The proposed software model (TBox); color-coded by defining ontology.

DUL is a restriction of DOLCE, which is why we follow the form of CSO in defining algorithms, implementing source code, and containing files. Concretely, we, too, distinguish between information content, structure, and physical manifestation as `Algorithm` (a DUL `Plan`), `ComputerProgram` (a `StructuredText`), and `ComputerFile` (a DUL `InformationRealization`). We also describe which language gives meaning to a text; in the case of a `ComputerProgram`, this is a `FormalLanguage`, but other pairs are possible, e.g., a poem written in a natural language or a picture encoded in some file format.

As explained in Section 3.4, however, we need to deviate from common definitions of software and running software instances. `Software` is for us a twofold `Design`. First, it describes some `SoftwareConfiguration`, a structured collection of source code and data. Second, it describes the structure, behavior, and function of any execution of itself. Abstracting to an intentional stance [57], we see the running instance of software as an `Agent`. This view enables us to describe the interactions of `SoftwareAgents` via SOMA's action-centric approach and the new taxonomies of communication and mental tasks, setting MOI apart from existing software ontologies (cf. Section 3.4).

A common requirement for agency is intentionality [58], which DUL also demands. However, in DUL, an agent may bestow intentionality upon others, e.g., in the case of an organization, it stems from its members, whom the organization `actsThrough`. In our case, some agent (a user, method caller, etc.) `actsThrough` the `SoftwareAgent`. We can further exploit this to model that some software agent depends on other software or on hardware. For example, a script that `actsThrough` some interpreter or a bytecode that `actsThrough` the computer processor represents the former's reliance on the latter.

Software categories are abundant. Instead of associating software with specific types, we argue that software agents can play different roles in different scenarios. A server is sometimes a client; a database can be used for arithmetic. MOI is meant to describe software via the ability to perform tasks, i.e., capabilities, and via played roles.

4.5. Capabilities

For CQ9, the term `Capability` typically describes what actions are expected of an agent. We follow Smith's view that a `Capability` is a `Disposition` [59,60]. As explained in Section 3.3, we disagree with defining it as yielding benefits to its bearer. Rather, we see it as the tendency of its bearer to actively participate in the associated `Task` as a `Performer`, and introduce the relation `affordsPerformer`. A `Capability`'s

bearer is always a performer. This generalizes SOMA's dispositional model to agents and allows us to develop taxonomies of mental tasks linked to complementary capabilities. The pattern was depicted in Figure 2. As an example, consider Figure 1: the roles `Sender` and `Receiver` are subclasses of `Performer`; the respective `SoftwareInstances` can be associated with matching `Capabilities` `CanSend` and `CanReceive`.

We further extend this by introducing part-hood for dispositions and capabilities. Utilizing DUL's `hasPart`, we decompose complex capabilities into simpler ones, e.g., the capability to answer a query has parts comprehending the query, selecting, encoding, and sending an answer. Since a `Disposition` is a `Quality` of an object in SOMA, we can concretize these further – e.g., the capability to reason is composed of qualities such as reasoning speed or soundness and completeness of the underlying logical formalism. Such information is essential to dynamically plan reasoning flows.

We can also model patterns from software engineering with capabilities: e.g., the requirement that plugins for a host software must implement its interface means they must have matching capabilities and dispositions. Sender and receiver must have the capability to understand the same language; code with the disposition to be interpreted needs an executor with a matching capability to understand the used programming language.

5. Evaluation

We briefly discuss MOI's reasoning properties. Using Konclude [61] in a virtual machine supported by a i7-1065G7 CPU, 32GB RAM, and a GTX 1660 Ti graphics card, initial classification of the current SOMA version, including MOI, takes about 3s, and subsequent DL queries about 60ms each. Since most axioms of MOI fall under the OWL QL profile, we believe that reasoning over large databases is tractable, as is already the case with non-MOI NEEMs in KnowRob. Note, however, that SOMA fully exploits the expressivity of OWL DL, e.g., for reasoning about capabilities. Tools like module extraction [62] help in selecting task appropriate subsets.

We hybridly evaluate MOI by using the CQs raised in Section 2 and by giving reasoning and querying examples. Following the SABiO guidelines [26], we constructed test-cases as formal queries associated with CQs, and executed these over example ABox-data with MOI as the reasoning schema. Since recalling metacognitive experiences means querying potentially massive amounts of data with complex constraints, DL queries or conjunctive queries might not be sufficient; instead, we chose SPARQL [63].

```
SELECT ?x ?y WHERE {?x causes ?y}
```

Figure 8. Query A, associated with CQ1

Figure 8 depicts the simple query A constructed from CQ1 to transitively return all pairs of causally related actions (due to space constraints, we drop namespaces). Scenarios include hunting error causes or reflecting on the consequences of past actions, e.g., when planning what to do next. Figure 9 contains a more complex example, in which we query for all actions that transitively contain (at least) two causally independent actions.

Query C (Figure 10) returns all `IllocutionaryTasks` together with their classified actions, roles played, and action participants. Subsequent queries might return the most specific concepts an individual satisfies, e.g., that a returned role is a `Sender`. SPARQL-

```
SELECT ?x WHERE {
  ?x hasPart ?y;
     hasPart ?z.
  MINUS {?y causes |
            isReactionTo ?z}
  FILTER (?y != ?z) }
```

Figure 9. Query B, associated with CQ1

```
SELECT ?act ?task ?part ?role WHERE {
  ?task a           IllocutionaryTask;
        isTaskOf    ?role;
        classifies ?act.
  ?part isParticipantIn ?act;
        hasRole         ?role}
```

Figure 10. Query C, associated with CQ2

queries associated with CQ4 might be similar. Due to space constraints, we do not show queries for CQ3, CQ7, and CQ8, whose patterns are already depicted in Figures 4 and 7.

The property chain reasoning for CQ5 can be exploited by query D from Figure 11. One use case for this is in finding errors/potential problems in inference sequences when a used premise is retrospectively found to be incorrect. Figure 12 shows query E associated with CQ6 that returns pairs of planning and execution tasks. MOI's capabilities are a simple extension of SOMA's dispositions, for which an algorithm to check matches has already been given [23]. The algorithm can be directly transferred to answer CQ9.

```
SELECT ?x ?y WHERE {
  ?x derivedFrom ?y}
```

Figure 11. Query D, associated with CQ4

```
SELECT ?task ?pTask WHERE {
  ?action executesTask       ?task.
  ?plan   definesTask        ?task.
  ?infObj expresses          ?plan.
  ?infRea realizes           ?infObj;
          hasRole            ?role.
  ?role   isCreatedOutputOf  ?pTask}
```

Figure 12. Query E, associated with CQ6

6. Conclusion and Future Work

We have proposed the *Meta-Ontology for Introspection (MOI)*, a novel, formal model to capture the metacognitive experiences of (robotic) CAs that integrates well into the established SOMA ontology. Following a formal ontology development and evaluation protocol, and building upon an elaborate survey of existing models, MOI is based on state-of-the-art research. We have shown that the model can trace telemetry between (software) agents and component-internal information processing, classify mental tasks, and model capabilities.

Note that since MOI's scope is limited to computational CAs, it may not be straightforward to transfer MOI to human cognition or human-CA interaction. Moreover, as is typical of early, exploratory work, further evaluation in application is necessary.

In the future, we aim to implement the model in CRAM and KnowRob as a framework for metacognitive monitoring, which any metacognitive control routines can then use. Further extensions should cover the properties of information objects, e.g., correctness, and the state of an agent's beliefs, desires, and intentions. When modeling concrete CAs, additional detail might be introduced to model mental and communication tasks.

Acknowledgement

The research reported in this paper has been supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320

Project-ID 329551904 “EASE - Everyday Activity Science and Engineering”, University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject “P05-N – Principles of Metareasoning for Everyday Activities” and by the FET-Open Project #951846 ”MUHAI – Meaning and Understanding for Human-centric AI” by the EU Pathfinder and Horizon 2020 Program.

References

[1] Wells A. Metacognitive therapy for anxiety and depression. Guilford press; 2011.
[2] Nelson TO, Narens L. Metamemory: A Theoretical Framework and New Findings. Psychology of Learning and Motivation. 1990;26:125-73.
[3] Flavell JH. Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry. American Psychologist. 1979;34:906-11.
[4] Kotseruba I, Tsotsos JK. 40 years of cognitive architectures: core cognitive abilities and practical applications. Artificial Intelligence Review. 2020;53(1):17-94.
[5] Laird J, Lebiere C, Rosenbloom P. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. AI Magazine. 2017 12;38:13-26.
[6] Kelley TD, Thomson R, Milton JR. Standard Model of Mind: Episodic Memory. In: Proc. of BICA. vol. 145 of Procedia Comput. Sci. Elsevier; 2018. p. 717-23.
[7] Sandini G, Sciutti A, Vernon D. Cognitive Robotics. In: Ency. of Robotics. Springer; 2021. p. 1-7.
[8] Beetz M, Mösenlechner L, Tenorth M. CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments. In: Proc. of IROS. IEEE; 2010. p. 1012-7.
[9] Beetz M, Beßler D, Haidu A, Pomarlan M, Bozcuoğlu AK, Bartels G. KnowRob 2.0 — A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents. In: Proc. of ICRA; 2018. p. 512-9.
[10] Beetz M, Tenorth M, Winkler J. Open-EASE: A Knowledge Processing Service for Robots and Robotics/AI Researchers. In: Proc. of ICRA; 2015. p. 1983-90.
[11] Alt B, Katic D, Jäkel R, Bozcuoglu AK, Beetz M. Robot Program Parameter Inference via Differentiable Shadow Program Inversion. In: Proc. of ICRA. IEEE; 2021. p. 4672-8.
[12] Cavallo A, Costanzo M, De Maria G, Natale C, Pirozzi S, Stelter S, et al. Robotic Clerks: Autonomous Shelf Refilling. In: Robotics for Intralogistics in Supermarkets and Retail Stores. Springer; 2022. p. 137-70.
[13] Salinas Pinacho L, Wich A, Yazdani F, Beetz M. Acquiring Knowledge of Object Arrangements from Human Examples for Household Robots. In: Proc. of KI. vol. 11117 of LNCS. Springer; 2018. p. 131-8.
[14] Bozcuoğlu AK, Kazhoyan G, Furuta Y, Stelter S, Beetz M, Okada K, et al. The Exchange of Knowledge Using Cloud Robotics. IEEE Robotics and Automation Letters. 2018;3(2):1072-9.
[15] Olivares-Alarcos A, Beßler D, Khamis A, Goncalves P, Habib MK, Bermejo-Alonso J, et al. A Review and Comparison of Ontology-based Approaches to Robot Autonomy. KER. 2019;34.
[16] Manzoor S, Rocha Y, Joo SH, Bae SH, Kim EJ, Joo KJ, et al. Ontology-Based Knowledge Representation in Robotic Systems: A Survey Oriented toward Applications. Applied Sciences. 2021;11:1-30.
[17] Koralewski S, Kazhoyan G, Beetz M. Self-Specialization of General Robot Plans Based on Experience. IEEE Robotics and Automation Letters. 2019;4(4):3766-73.
[18] Metzler T, Shea K. Taxonomy of cognitive functions. In: Proc. of ICED; 2011. p. 330-41.
[19] IEEE Robotics and Automation Society. IEEE Standard Ontologies For Robotics and Automation; 2015.
[20] Pignaton de Freitas E, Olszewska JI, Carbonera JL, Fiorini SR, Khamis A, Ragavan SV, et al. Ontological Concepts for Information Sharing in Cloud Robotics. JAIHC. 2020 06:1-12.
[21] Beßler D, Porzel R, Pomarlan M, Vyas A, Höffner S, Beetz M, et al. Foundations of the Socio-Physical Model of Activities (SOMA) for Autonomous Robotic Agents. In: Proc. of FOIS; 2021. p. 159-74.
[22] Waibel M, Beetz M, Civera J, D’Andrea R, Elfring J, Gálvez-López D, et al. RoboEarth. IEEE Robotics & Automation Magazine. 2011;18(2):69-82.
[23] Beßler D, Porzel R, Pomarlan M, Beetz M, Malaka R, Bateman J. A Formal Model of Affordances for Flexible Robotic Task Execution. In: Proc. of ECAI; 2020. p. 2425-32.
[24] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A. Wonderweb deliverable d18; 2003. Tech. rep.

[25] Gangemi A. The DOLCE+DnS Ultralite ontology (DUL) Version 4.0; 2021. Available from: http://www.ontologydesignpatterns.org/ont/dul/DUL.owl.
[26] de Almeida Falbo R. SABiO: Systematic Approach for Building Ontologies. In: Proc. of ONTO.COM/ODISE; 2014. p. 1-14.
[27] Grüninger M, Fox MS. Methodology for the Design and Evaluation of Ontologies. In: Proc. of the WS on Basic Ontological Issues in Knowledge Sharing; @IJCAI 1995. p. 1-10.
[28] Poldrack R, Yarkoni T. From Brain Maps to Cognitive Ontologies: Informatics and the Search for Mental Structure. Annual review of psychology. 2015 09;67.
[29] Poldrack RA, Kittur A, Kalar D, Miller E, Seppa C, Gil Y, et al. The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. Frontiers in neuroinformatics. 2011;5:17.
[30] Blanch A, Garcia R, Planes J, Gil R, Balada F, Blanco E, et al. Ontologies about human behavior. European Psychologist. 2017.
[31] Caro M, Cox M, Toscano R. A Validated Ontology for Metareasoning in Intelligent Systems. Journal of Intelligence. 2022 11;10:113.
[32] Chang DS, Cho GH, Choi YS. Ontology-Based Knowledge Model for Human-Robot Interactive Services. In: Proc. of SAC; 2020. p. 2029–2038.
[33] Hernández Corbato C. Model-based Self-awareness Patterns for Autonomy [Phd Thesis]. Universidad Politécnica de Madrid; 2014.
[34] Hernández Corbato C, Milosevic Z, Olivares C, Rodriguez G, Rossi C. Meta-control and Self-Awareness for the UX-1 Autonomous Underwater Robot. In: Proc. of ROBOT; 2019. p. 404-15.
[35] Schmill MD, Anderson ML, Fults S, Josyula D, Oates T, Perlis D, et al. The Metacognitive Loop and Reasoning about Anomalies. In: Metareasoning: Thinking about Thinking. The MIT Press; 2011. p. 183-98.
[36] Madera-Doval DP. A Validated Ontology for Meta-Level Control Domain. Acta Sci Inform. 2019;3(3).
[37] El Asmar B, Chelly S, Färber M. AWARE: An Ontology for Situational Awareness of Autonomous Vehicles in Manufacturing. In: Proc. of CSKG; 2021. p. 1-12.
[38] Vacura M. Modeling Artificial Agents' Actions in Context – a Deontic Cognitive Event Ontology. Appl Ontol. 2020;15(4):493–527.
[39] Otte JN, Beverley J, Ruttenberg A. BFO: Basic Formal Ontology. Appl Ontol. 2022;17(1):17-43.
[40] Merrell E, Limbaugh D, Anderson A, Smith B. Mental capabilities. In: Proc. of ICBO; 2019. p. 1-7.
[41] Kunze L, Roehm T, Beetz M. Towards semantic robot description languages. In: Proc. of ICRA; 2011. p. 5589-95.
[42] Wang X, Guarino N, Guizzardi G, Mylopoulos J. Towards an ontology of software: a requirements engineering perspective. In: Proc. of FOIS; 2014. p. 317-43.
[43] Oberle D. Semantic Management of Middleware. vol. 1 of Semantic Web and Beyond. Springer; 2006.
[44] Lando P, Lapujade A, Kassel G, Fürst F. An Ontological Investigation in the Field of Computer Programs. In: Software and Data Technologies. Springer; 2007. p. 371-83.
[45] Duarte B, de Castro Leal A, de Almeida Falbo R, Guizzardi G, Guizzardi R, Souza V. Ontological foundations for software requirements with a focus on requirements at runtime. Appl Ontol. 2018;13(2):73-105.
[46] de Oliveira JMP. An Ontological Analysis From Algorithm to Computer Capability. In: Proc. of ONTOBRAS; 2020. p. 48-60.
[47] Mizoguchi R, Borgo S. YAMATO: Yet-another more advanced top-level ontology. Appl ontol. 2022;17(1):211-32.
[48] Hitzler P, Krötzsch M, Parisa B, Patel-Schneider F, Rudolph S, editors. OWL 2 Web Ontology Language Primer (Second Edition); 2012. Available from: www.w3.org/TR/owl2-syntax.
[49] Herman D. Narrative Theory and the Cognitive Sciences. Narrative Inquiry. 2003 01;11:1-34.
[50] Francken JC, Slors M. From commonsense to science, and back: The use of cognitive concepts in neuroscience. Consciousness and Cognition. 2014;29:248-58.
[51] Fan J, Barker K, Porter B, Clark P. Representing Roles and Purpose. In: Proc. of K-CAP; 2001. p. 38–43.
[52] Austin JL. How to do things with words. Oxford university press; 1975.
[53] Shannon CE, Weaver W. The mathematical theory of communication. University of Illinois Press; 1949.
[54] Nanay B. Motor Imagery and Action Execution. Ergo. 2020 05;7.
[55] Rosenbaum DA. Human motor control. Academic press; 2009.
[56] Ye H, Zeng H, Yang W. Enactive cognition: Theoretical rationale and practical approach. Acta Psycho-

logica Sinica. 2019;51(11):1270-80.
[57] Dennett DC. The intentional stance. MIT press; 1987.
[58] Schlosser M. Agency. In: Zalta E, editor. The Stanford Encyclopedia of Philosophy; Winter 2019. p. 1.
[59] Smith B. The Ontology of Capabilities; 2019. Slides presented in an IOF Meeting.
[60] Merrell E, Limbaugh D, Koch P, Smith B. Capabilities; 2022. Available from: https://philarchive.org/archive/MERC-14.
[61] Steigmiller A, Liebig T, Glimm B. Konclude: system description. J Web Semant. 2014;27:78-85.
[62] Grau BC, Horrocks I, Kazakov Y, Sattler U. Modular reuse of ontologies: Theory and practice. JAIR. 2008;31:273-318.
[63] W3C SPARQL Working Group, editor. SPARQL 1.1 Overview; 2013. Available from: https://www.w3.org/TR/sparql11-overview/.