

exploit affordances in the system based on the set of behaviors $B = \{b_1, b_2, \dots, b_n\}$, e.g., $B = \{gripper-closable, login, \dots\}$.

We define a *signifier* sig as a perceivable cue or sign that can be interpreted meaningfully to reveal information about an affordance aff_1 , formally defined as follows:

$$sig = (sp_b, A, c, salience), \quad (5)$$

where:

- sp_b is the signified *specification* of a behavior b of the set B , i.e. the specification of a course of action that can be performed by an agent when aff_1 is exploitable;
- A is a set of *abilities*, where an ability is a quality of an agent to perform a behavior of the set B . Hence, sig recommends that agents should have all the abilities of the set A so that aff_1 becomes exploitable.
- c is the *context* which sig recommends to hold so that aff_1 becomes exploitable, i.e. constraints to which the agent-environment situation is recommended to conform;
- *salience* is the quality of sig that indicates how useful or relevant sig is.

Lst. 1 (l.7-13) captures a machine-understandable signifier in RDF [20] that follows Def. 5: It signifies the specification of the behavior `close-gripper`, recommended to be performed when an agent is able to become an operator and the gripper is open.

Note that the definition (and the representation) of a signifier (here, sig) does not include any direct reference to an affordance (here, aff_1). This is preferable for two reasons: First, affordances emerge upon the presence of *individual agents* in the appropriate situation, while signifiers are defined with respect to *agent types*. This is essential because designers typically do not have prior knowledge of the individual agents that will inhabit the environment. In this way, Def. 5 preserves the evolvability and reusability of signifiers, since agents *remain more loosely coupled* to their environment. Second, *affordances* are defined in Affordance Theory for studying how *animals* control their behaviors. However, we model *signifiers* for supporting interactions of *autonomous agents*, which exhibit more heterogeneous cognitive and sensorimotor abilities than animals – including agents whose cognitive abilities heavily rely on computational processes such as reasoning over representations, and whose sensorimotor abilities⁵ allow for behaviors that are much broader than animal behaviors.

Even though there is no formal relation between signifiers and affordances, the designer of a signifier sig may use the elements of Def. 5 to establish a (direct or indirect) relationship to specific elements of aff_1 . The forms that this relationship can take can be narrowed down to the following cases:

- (C1) sp_b of the behavior b is equivalent to the specification of the behavior b_1 for exploiting aff_1 and optionally any behaviors that lead to making aff_1 exploitable, i.e. sp_b specifies behaviors of the set $B' \subseteq B$, and $B' \supseteq \{b_1\}$;
- (C2) A is equivalent to the set of any of the abilities that lead to making aff_1 exploitable, i.e. $A \subseteq Ability-to-B$;

- (C3) c is equivalent to (part of) the constraints that are satisfied in situation s_1 .

C1-C3 should be used based on the experience and expectations that environment designers hold about targeted agents towards enabling effective and efficient interactions. Therefore, designers are responsible for choosing when to aggregate or omit to refer to information about the sets *Affords-B* and *Ability-to-B*. For example, (C1) allows for managing the granularity of behavior b that is specified in the signified specification sp_b . Considering that aff_1 is the affordance `gripper-closable`, then b could be specified as the sequence $\langle b_2, b_1 \rangle = \langle login, close-gripper \rangle$. Such granularity could be desirable if b_2 and b_1 are expected to be frequently performed in sequence or for enabling agents to access specifications of higher-level behaviors. Similarly, (C2) enables designers to avoid including in A these abilities of *Ability-to-B* that targeted agents are expected to always have. (C3) enables recommending a context c by determining which constraints address the most significant aspects of situation s_1 that makes aff_1 exploitable. Generally, the context may capture constraints about agents and constraints about the entity offering an affordance, which are disjoint (e.g., constraints on the intentions of the agent, and respectively the state of the artifact) or interdependent (e.g., that the agent is situated in the same working environment that contains the environmental).

In our proposed model of signifiers, recommended abilities and context are not meant to *regiment* agents' behaviors, i.e. to constrain whether an agent will actually exploit an affordance. However, they can both be beneficial in evaluating the relevance of an affordance exploitation, and consequently the relevance of the signifier for an agent with given abilities in a given situation.

We further develop our model to define how *behaviors* can be specified for *hypermedia-driven interactions* in Hypermedia MAS, e.g., in the form of an AgentSpeak plan [4], a JADE behavior [3], etc. Due to this diversity, the only requirement that we formally impose for the definition of a behavior specification is that it specifies at least one action.

Listing 1: A (generic) signifier that reveals information about the affordance `gripper-closable` of a robotic arm.

```

1 @base <http://ex.org/wksp/1/arts/1>.
2 @prefix hctl:<https://www.w3.org/2019/wot/hypermedia#>.
3 ...
4 <#sig> a hmas:Signifier ;
5   hint:signifies <#close-gripper> ;
6   hint:recommendsAbility [ a manu:OperatorAbility ] ;
7   hint:recommendsContext <#env-context> .
8
9 <#env-context> a hint:Context; sh:targetNode ex:leubot ;
10  sh:property [ sh:path manu:hasGripperValue ;
11                sh:hasValue "500"^^xsd:integer ] .
12
13 <#close-gripper> a hint:ActionSpecification;
14  hint:hasForm [ hctl:hasTarget leubot:base ;
15                hctl:forContentType "application/json" ] ;
16  hint:expects [ a hint:Input;
17                hint:hasSchema <#gripper-schema> ] .
18
19 <#gripper-schema> a js:ObjectSchema ;
20  js:properties [ a js:IntegerSchema ;
21                 js:propertyName "manu:hasGripperValue"; js:enum "0"] .

```

⁵In Affordance Theory, action is considered to be strictly coupled to perception. Although there are examples of autonomous agents which exhibit similar cognitive and sensorimotor abilities for performing perception-to-action behaviors, such as Rodney Brooks' physically embodied agents that implement the subsumption architecture [5], a Hypermedia MAS is expected to feature a greater diversity of agents' abilities.

An *action* is specified through a) *forms* that describe the hypermedia controls that can be used to implement and execute the action and b) optionally an *input*. For example, the behavior `close-gripper` could be specified as an action through a form that describes an HTTP request, and a gripper input value. An *action execution* can be treated as a behavior that is expected to result in a modification of the state of the environment (e.g., of the gripper). In this case, any entities in the system that monitor an agent executing an action, may perceive that the agent acts on the environment. However, an action could also be used for perception in the case in which an agent executes the action with the purpose of affecting its perception. Then, a specification sp_b formally specifies such a behavior b (an action execution) as follows⁶:

$$sp_b = (Forms, [Input]), \quad (6)$$

where *Forms* is a set of forms where each form describes an implementation of the specified action, and *Input* is the input expected when it is possible or required to parameterize the action execution. List. 1 (l.15-24) captures a signified specification of `close-gripper`.

3.1.2 Abilities of Autonomous Agents. In the following, we discuss how *abilities* recommended by signifiers can facilitate the effective and efficient affordance discovery and exploitation in the MAS. For example, an agent reasoning on signifiers could infer that it should exploit a related affordance only if it has the appropriate abilities. Even if the agent does not currently exhibit the recommended abilities, it could still use this information towards extending its abilities, or delegating goals to capable agents. For example, if an agent ag_1 has the abilities of the set A_1 and is aware of a signifier $sig_1 = (sp_b, A, c, salience)$, where $A \not\subseteq A_1$, then ag_1 may decide to forward sig_1 and request the performance of the related behavior from an agent ag_2 that has the abilities of the set A_2 , where $A \subseteq A_2$. Next, we consider the abilities of agents to illustrate diverse aspects of agents that signifier designers could take into account.

An agent may have an *ability to reason on a representation of its internal state*; e.g., a BDI agent can reason about its beliefs, desires, and intentions [26]. Recommending such abilities ensures that agents have the appropriate *cognitive skills* to perform a specified behavior. For instance, a signifier that signifies the specification of sending a message with mentalistic semantics (e.g., in KQML [18]) in the context of an interaction protocol should recommend the ability of an agent to have mental states, in order to support compliance to the protocol.

An agent may, further, have an *ability to reason about actions and to plan ahead by using specific methods and mechanisms*, e.g., by using a STRIPS-like planner to synthesize plans [12]. Then, a signifier that targets agents with such a planning ability should signify behavior specifications that can contribute to the construction of a suitable planning domain (e.g., specifications of actions with their preconditions and postconditions). Alternatively, a signifier that targets BDI agents that implement the Procedural Reasoning System [12] could instead signify behavior specifications as plans.

An agent also may have an *ability to behave based on its social context*, e.g., to fulfill its role of machine operator by operating machines in an industrial shopfloor. Recommending role-related abilities within a multi-agent setting (e.g., an organization [32]) can

enable agents to interpret which affordances are exploitable based on a role or help to fulfill a role. For example, a signifier specifying a behavior for operating a robotic arm on a manufacturing shopfloor should recommend the ability of an agent to play the role of a robot operator rather than the role of a warehouse manager.

An agent also may have an *ability to operate within a given domain*, e.g., by having knowledge of abstractions and processes in industrial manufacturing. For example, a signifier that signifies a behavior specification using a specific semantic model for industrial processes, such as the SAREF4INMA⁷ semantic model, should recommend the ability of an agent to interpret the required model.

An agent, finally, may have an *ability to perform a behavior when an affordance is present*, e.g., to behave based on a behavior specification. Such abilities could originate from the set *Ability-to-B* and their recommendation by a signifier falls under case (C2). For example, a signifier that signifies a specification for `close-gripper` (Lst. 1) could recommend the ability of an agent to *log in* as an operator.

3.2 Environment Support for Signifier Exposure

Based on our formalization of signifiers, and the discussion of the diverse types of agent abilities that can be considered with our proposed model, we next discuss how the *exposure* of signifiers can be realized and dynamically adjusted in hypermedia environments.

3.2.1 Exposing Signifiers in the Environment. Signifiers are not simple informational resources, but rather constructs available in hypermedia environments that enable *situated* agents to discover and interpret affordances. Here, we consider that signifiers can be exposed through *workspaces* that logically partition the environment (similar to A&A workspaces [27])⁸.

Specifically, a workspace in a hypermedia environment is a container of Web resources and the interactions that are enacted among contained Web resources. A workspace w is formally defined as:

$$w = (R, I), \quad (7)$$

where R is the set of resources contained in the workspace w (e.g., agents, signifiers, other workspaces etc.) and I is the set of interactions that take place in w , enacted among resources of the set $IR \subseteq R$ that offer and exploit affordances.

We further define that $IR = Ag \cup Ar$, where Ag is the set of the agents that are situated in the workspace w , and Ar is the set of the non-autonomous entities, i.e. artifacts [27], that are contained in w . Then, signifiers are designed to enable the interactions of resources in IR , i.e. offered by artifacts in Ar to agents in Ag . To examine what the environment affords in the context of agent-to-agent interaction, an affordance offered by an agent $ag \in Ag$ is treated as an affordance of the body⁹ $body_{ag}$ of ag , where $body_{ag} \in Ar$ ¹⁰.

⁷SAREF4INMA defines a vocabulary for the industry and manufacturing domain; available online: <https://saref.etsi.org/saref4inma/v1.1.2/>

⁸We formalize workspace to provide environment support for managing interaction cues/metadata through containers (e.g. as in HCI via the “natural mapping” principle [24], or in Web-based systems via W3C directories [30] and API ecosystems [23]). However, we do not expect (or impose) that workspaces are the only means to manage MAS environments and signifier exposure.

⁹The body of an agent is an artifact that enables the agent to be situated in a workspace and to interact with other resources in the workspace [28].

¹⁰Following the definition of affordances (cf. Def. 1), we only consider agent-to-environment interaction. Agent-to-agent interaction is managed as agent-to-environment interaction through agent bodies, and environment-to-environment (artifact-to-artifact) interaction is not considered.

⁶We insert [and] to delimit expressions considered optional.

However, the work of designers should not be limited to the construction of signifier tuples as given by Def. 5. Instead, upon enriching the environment with signifiers, designers should in addition consider how to support the discoverability of signifiers by agents that are expected in the environment *at run time*.¹¹ To advertise signifiers for affordances exposed by an *artifact*, the environment designer can create an *artifact profile*, i.e. structured data describing the artifact through signifiers and general (domain- and application-specific) metadata. Formally, a profile p_{ar} of an artifact ar is defined as the following construct:

$$p_{ar} = (Sig_{ar}, s_{ar}), p_{ar} \in R, \quad (8)$$

where Sig_{ar} is the set of the signifiers that are exposed in the profile p_{ar} , and s_{ar} is metadata capturing part of the situation of the ar .

For an artifact ar in a workspace w , at least one artifact profile p_{ar} is contained in w to explicitly capture the containment relationship between ar and w . Then, the discovery of the signifier set Sig_{ar} in p_{ar} is enabled by the containment of ar in w .

3.2.2 Dynamically Adjusting Signifier Exposure. In large-scale environments, additional support for the discoverability of signifiers is needed as the number of artifacts, affordances and, consequently, signifiers grows. At the same time, the interpretability of signifiers may be hindered within open environments where agents interpret signifiers based on diverse abilities.

To address these issues we propose a *Signifier Exposure Mechanism* (SEM) (cf. [21]) that exposes a filtered set of signifiers based on the targeted agent's abilities and situation (e.g., the agent's goals), and the situations of artifacts (e.g., based on the valid transitions from the current artifact state).

First, we consider the set P_{Ag} that is the set that contains the profiles of each agent $ag \in Ag$ in the workspace w . Then, each profile $p_{ag} \in P_{Ag}$ is metadata describing the abilities and the situation of the agent ag . The profile p_{ag} of the agent ag is then formally defined as the following construct:

$$p_{ag} = (A_{ag}, s_{ag}), p_{ag} \in P_{Ag} \subset R, \quad (9)$$

where A_{ag} is the set of the abilities of the agent ag , and s_{ag} is metadata capturing part of the situation of the agent ag .

Additionally, we consider the set P_{Ar} that is the set that contains the profiles of each artifact $ar \in Ar$ (cf. Def. 8) in the workspace w .

We now present a definition of an SEM whose functionality is described as follows: Given an agent profile $p_{ag} \in P_{Ag}$, and given an artifact profile $p_{ar} \in P_{Ar}$, the SEM outputs an artifact profile p'_{ar} that exposes only those signifiers of p_{ar} that match a) the abilities of agent ag and b) the agent-environment situation (i.e. the ag - ar situation). For the former, we simply consider if the set of recommended abilities is a subset of the agent's abilities. For the latter, we use an evaluation function E that evaluates to what degree the situation of the agent and the situation of the artifact conform to the context that is recommended by the signifier. For this, we consider C_{ar} as the set of all the recommended contexts that can be accessed through the profile of artifact ar . Then, we use SC_{Ar} to denote the set of all the recommended contexts that can be accessed through the set of artifact profiles P_{Ar} , paired with the

¹¹This represents a significant broadening of the signifier concept with respect to the HCI literature, where signifiers are thought as being able to be modulated *at run time* only in edge cases.

situations that are captured in their corresponding profiles of P_{Ar} . Pairs $(c_{ar}, s_{ar}) \in SC_{Ar}$ are used for associating the situation of an artifact ar only with the contexts that apply to the same artifact ar :

$$SC_{Ar} = \{C_{ar} \times \{s_{ar}\} \mid p_{ar} = (Sig_{ar}, s_{ar}) \wedge ar \in Ar\}. \quad (10)$$

We, also, use S_{Ag} to denote the set that includes the situations that can be accessed through the set of agent profiles P_{Ag} , formally:

$$S_{Ag} = \{s_{ag} \mid p_{ag} = (A_{ag}, s_{ag}) \wedge ag \in Ag\} \quad (11)$$

Then, we define the evaluation function E that validates an agent situation of the set S_{Ag} and an artifact situation of the set CS_{Ar} against a context of the set CS_{Ar} :

$$E : CS_{Ar} \times S_{Ag} \longrightarrow [0, 1]. \quad (12)$$

Finally, P'_{Ar} is the set of possible produced artifact profiles:

$$P'_{Ar} = \{(Sig'_{ar}, s_{ar}) \mid p_{ar} = (Sig_{ar}, s_{ar}) \wedge Sig'_{ar} \in \mathcal{P}(Sig_{ar}) \wedge ar \in Ar\} \quad (13)$$

We can now formally define SEM as follows:

$$\begin{aligned} SEM : P_{Ag} \times P_{Ar} \times \{e \mid e \in [0, 1]\} &\longrightarrow P'_{Ar}; \\ ((Sig_{ar}, s_{ar}), (A_{ag}, s_{ag}), t) &\longmapsto (\{sig = (sp_b, A, c, salience) \in Sig_{ar} \\ &\mid A \subseteq A_{ag} \wedge E((c, s_{ar}), s_{ag}) > t\}, s_{ar}) \end{aligned} \quad (14)$$

where t is a threshold value that is given as input to the SEM.

While it may be beneficial, it is not required that exposed signifiers should always relate to affordances that are considered exploitable or relevant. For example, there may be cases where observations about agents and artifacts are insufficient for evaluating relevance, for instance within highly dynamic environments or when an agent prefers not to share such information. Additionally, agents with planning abilities may require an action space that relates to affordances which are not necessary currently exploitable.

3.3 Customizing Signifiers to Heterogeneous Reasoning Abilities of Agents

In this section, we examine how a generic signifier can be extended to reveal information that is relevant to agents with different reasoning and planning abilities. We, specifically, consider the design of signifiers for a) BDI agents that implement the Procedural Reasoning System (PRS), and b) agents capable of planning their actions by using a STRIPS-like planner.

To illustrate our approach with the generic signifier in Lst. 1, which signifies an action specification for closing the gripper of a robotic arm. In the following, we show how this signifier can be extended to customize its content towards accommodating the abilities of the targeted agents.

3.3.1 Customizing Signifiers for PRS-based Reasoning. A BDI agent with a PRS-based ability is expected to look for signifiers for satisfying its *intentions*, i.e. the *goals* that the agent has committed to achieve. Although the agent's *plan library* may already contain a plan, i.e. a course of action for accomplishing the agent's goal, a dynamic environment does not always permit for the required actions to be coupled to specific action implementations. More concretely, in case an agent has access to a pick-and-place plan, it may intent to act on a robotic arm (e.g., to close an arm's gripper), but without

knowing which robotic arm to use in its current workspace, or any other lower-level interaction details.

To discover fitting signifiers in our approach, an agent may update its profile to describe (part of) its *mental attitudes* – its current *beliefs* and *goals*, for instance, its desire to achieve a specific goal while looking for the appropriate means to satisfy its desire. Considering the desire of an agent to achieve the goal of picking up an item and placing it in a specified location, the agent’s profile could be similar to the profile presented in Lst. 2¹².

Taking into account the abilities of PRS-based agents, designers can now construct signifiers recommending contexts that relate to the expected situations of the targeted agents, i.e. situations in terms of *beliefs*, *desires*, and *intentions*. Lst. 3 is an example of a signifier for the affordance gripper-closable of a robotic arm, which *extends*

¹²We consider that the agent has a desire that is a goal achievement as specified in the AgentSpeak language [4].

Listing 2: The resource profile of a BDI agent that implements the PRS and desires to pick and place an item.

```

1 @base <http://ex.org/wksp/1/arts/2>.
2 @prefix item-profile: <http://ex.org/wksp/1/arts/3#>
3 ...
11 <> a hmas:AgentProfile ; hmas:isProfileOf <#agent> .
12
13 <#agent> a hmas:Agent ;
14 hint:hasAbility [ a prs:PRSAbility ] ;
15 hint:hasAbility [ a manu:OperatorAbility ] ;
16 prs:hasDesire [ a prs:GoalAchievement,manu:PickAndPlace;
17 prs:hasInputList [ a rdf:List ;
18 rdf:first wksp:item ;
19 rdf:rest [ a rdf:List ;
20 rdf:first <#location> ] ] .
21
22 item-profile:item a manu:Item ;
23 manu:hasLocation item-profile:location .

```

Listing 3: A customized signifier for agents that implement a BDI architecture based on the PRS.

```

1 ...
9 <#sig> a hmas:Signifier ;
10 hint:signifies <#close-gripper> ;
11 hint:recommendsAbility [ a prs:PRSAbility ] ;
12 hint:recommendsAbility [ manu:OperatorAbility ] ;
13 hint:recommendsContext <#env-context>, <#prs-context> .
14
15 <#prs-context> a hint:Context; sh:targetClass hmas:Agent ;
16 sh:property [ sh:path prs:hasDesire ;
17 sh:minCount 1 ; sh:qualifiedMinCount 1 ;
18 sh:qualifiedValueShape <#desire-shape> ] .
19
20 <#desire-shape> a sh:NodeShape ;
21 sh:class manu:PickAndPlace;
22 sh:property [ sh:path prs:hasInputList
43 ... ] .
44
45 <#item-shape> a sh:NodeShape ;
46 sh:class manu:Item ;
47 sh:property [ sh:path manu:hasLocation ;
59 ... ] .
60
61 <#location-shape> a sh:NodeShape ;
62 sh:class manu:Location ;
63 sh:property [ sh:path manu:inRangeOf ;
64 sh:minCount 1 ;
65 sh:hasValue ex:leubot ] .

```

the generic signifier for the same affordance (Lst. 1) to better cater to such agents. In this case, the recommended context is a set of constraints that should be validated by the agent-environment situation in order for the signifier to be considered relevant for the agent. The constraints are expressed in the Shapes Constraint Language (SHACL)¹³ and concern the following aspects of the agent-artifact situation: 1) the agent has the desire to pick an item and place it in a target location (l. 20-43); 2) the item is in range of the robotic arm that offers the affordance gripper-closable (l. 45-59, 61-65); 3) the target location is in the range of the robotic arm that offers the affordance gripper-closable (l. 45-59, 61-65).

Additionally, the signifier is extended to recommend a *PRS-based ability*. Therefore, if an agent with such an ability looks up signifiers at run time, and an SEM is available (see Sect. 3.2), the SEM exposes the *customized* signifier of Lst. 3 rather than the generic signifier of Lst. 1. Finally, if the SEM has a SHACL validation feature, exposure is adjusted based on whether the agent-environment situation (Lst. 2) conforms to the SHACL context shape¹⁴. In this case, the customized signifiers are exposed to agents with a PRS-based ability and only if the situation conforms to the recommended constraints.

3.3.2 Customizing Signifiers for STRIPS-based Reasoning. Instead of profiting from signifiers that are designed with agents’ beliefs, desires, and intentions in mind, agents that are capable of automated planning would require signifiers that signify action specifications suitable to enrich a planning domain. For instance, Lst. 4 presents a customized signifier for the affordance gripper-closable of a robotic arm, which extends the generic signifier of Lst. 1 to signify an action specification that is relevant to agents with a STRIPS-based planning ability. The action specification specifies the type of the action that can be performed, and, additionally, the *preconditions*

¹³The SHACL specification is available online: <https://www.w3.org/TR/shacl/>

¹⁴An SEM with a SHACL processor has already been presented in [29]. There, agents specify SHACL shapes that impose constraints on signifiers, and they provide such shapes as input to the SEM upon looking for conforming signifiers. On the other hand, here, we consider that environment designers create the signifiers and specify constraints for identifying at run time conforming agent-environment situations.

Listing 4: A customized signifiers for agents with a STRIPS planning ability.

```

1 @prefix pddl:
2 <http://www.cs.yale.edu/homes/dvm/daml/pddlonto.daml#>.
3 ...
9 <#sig> a hmas:Signifier ;
10 hint:signifies <#close-gripper> ;
11 hint:recommendsAbility [
12 a strips:StripsPlanningAbility ] .
13
14 <#close-gripper> a hint:ActionSpecification;
15 ...
21 a a pddl:Action ;
22 pddl:action-label "closeGripper";
23 pddl:parameters [ a pddl:Param_seq ;
24 rdf:_1 <#param1> ] ;
25 pddl:precondition [
31 ... ] ;
32 pddl:effect [
41 ... ] .
42
43 <#param1> a pddl:Param ;
44 pddl:name "?gv" ;
45 drs:type manu:GripperValue ;
46 :hasSchema <#gripper-schema> .

```

and the *effects* of executing the action. For this example, we reused the PDDL ontology presented in [22] to specify actions that can be used to enrich a PDDL domain and, thus, become suitable input to a PDDL automated planner. The signifier is also extended to recommend a *STRIPS planning ability*, so that an SEM can adjust the signifier exposure based on the abilities of the requesting agent.

4 IMPLEMENTATION AND DEPLOYMENT

To ground our approach in a concrete use case, we present a prototypical Hypermedia MAS and a demonstrator scenario.

4.1 Prototype Deployment

In our implementation¹⁵, we use the Yggdrasil platform for Hypermedia MAS [8] as a repository for our hypermedia environment. We extended the Yggdrasil API to enable the publication of agent and artifact profiles as presented in Sect. 3.1. The API hence permits the publication and management of resource profiles based on a) the Hypermedia MAS Core Ontology (HMAS-core) that is used to describe core aspects of Hypermedia MAS (e.g., workspaces, resource profiles etc.), and b) the Hypermedia MAS Interaction ontology which extends HMAS-core based on our model of signifiers.

We further extended Yggdrasil with an SEM for dynamically adjusting the signifier exposure based on the formal model presented in Sect. 3.2. Specifically, agent and artifact profiles can be published so that they are accessible through workspaces of the hypermedia environment. As a result, when an agent discovers an artifact profile in the hypermedia environment, the SEM is responsible for evaluating which signifiers should be exposed to the agent: The SEM identifies which agent is currently looking for signifiers, and attempts to retrieve the agent’s profile. If no agent profile is available, the artifact profile is provided to the agent without undergoing any signifier adjustment. On the other hand, if the SEM acquires access to an agent profile, it proceeds to evaluate the complementarity between the agent and the artifact as given by Def. 14. For example, if the agent exhibits specific abilities, then the SEM will construct a variation of the artifact profile that exposes only these signifiers that contain recommendations for the specified agent abilities.

For our prototype, we used Jason [4] agents which were implemented and deployed in a JaCaMo¹⁶ application. The signifiers used were designed for revealing information about affordances of a PhantomX AX12 Reactor Robot Arm¹⁷.

4.2 Deployment Scenario

To validate our prototype implementation, we considered the case of a Hypermedia MAS where affordance discovery and exploitation take place in a workspace with industrial devices based on the following scenario: A manufacturing workspace contains a robotic arm that offers affordances to agents that are situated in the workspace, such as affordances to move the gripper and the base of the robotic arm. The robotic arm is modelled as an *artifact* and its presence is indicated in the hypermedia environment through its *artifact profile* that is contained in the workspace. Signifiers that

reveal information about the affordances of the robotic arm are exposed through its profile, and have been designed by taking into consideration the different abilities of agents that are expected to be situated in the manufacturing workspace. Specifically, we have considered three types of signifiers with regard to agents’ abilities:

- Signifiers that recommend the abilities of BDI agents that implement the Procedural Reasoning System (PRS).
- Signifiers that recommend the abilities of agents that can perform automated planning using a STRIPS-like planner.
- Generic signifiers that are not customized to agents with any specific abilities of reasoning about action and acting.

A BDI agent that implements the PRS joins the workspace with the objective to pick and place an item. The agent’s presence in the workspace is indicated in the hypermedia environment through its profile, i.e. metadata describing that a) the body of the agent is contained in the workspace, b) the agent desires to pick-and-place, and c) the agent has a PRS ability. The agent has already access to a relevant pick-and-place plan through its plan library, however, it is unaware of any lower-level implementation details that are required for effectively executing its plan. For this, the agent looks for signifiers to acquire information about the hypermedia controls that can be used to exploit relevant affordances. Upon looking for information about the affordances of the robotic arm that is contained in the manufacturing workspace, the agent receives only those signifiers that are relevant to its abilities from the SEM.

Another agent joins the workspace with the same objective of picking and placing an item. Although the agent does not currently have a relevant pick-and-place plan, it has access to a STRIPS-like planner for performing automated planning – an ability that is indicated in the agent’s profile. The agent decides to look for signifiers that may help it to synthesize a pick-and-place plan. Upon focusing on the robotic arm, the agent receives the profile of the robotic arm that exposes only signifiers that recommend a STRIPS planning ability. Consequently, the agent acquires access to action specifications that would be suitable for enriching a planning domain as they specify the types of actions that can be performed upon the robotic arm, along with action preconditions and effects.

5 CONCLUSIONS

In this paper, we introduced signifiers as a first-class abstraction in Hypermedia MAS. This was accomplished through formalizations for the design of signifiers that enable the hypermedia-driven exploitation of affordances on the Web, and that capture the agent-environment complementarity required to exploit affordances. These signifiers are coupled with a mechanism that describes the dynamic adjustment of signifier exposure based on the agent-environment situation. We provided examples and demonstrated how signifiers decouple behaviors from their implementations through hypermedia, and can be customized with respect to different reasoning abilities of agents, towards enabling the effective and efficient affordance discovery and exploitation in affordance-rich and open Web-based MAS. Based on the presented work, we aim at supporting and evaluating the effectiveness and efficiency of more complex behaviors through the design and exposure of signifiers based on the HATEOAS principle and the behavior-ability impredicativity of Affordance Theory.

¹⁵The implementation is available online: <https://github.com/Interactions-HSG/yggdrasil/tree/feature/sem>

¹⁶The JaCaMo documentation is available online: <https://jacamo.sourceforge.net/>

¹⁷<https://www.trossenrobotics.com/p/phantomx-ax-12-reactor-robot-arm.aspx>

ACKNOWLEDGMENTS

This research has received funding from the Swiss National Science Foundation under grant No. 189474 (*HyperAgents*) and from the European Union’s Horizon 2020 research and innovation program under grant No. 957218 (*IntelloIoT*). We thank Yousouf Taghzouti for his valuable input on the topic of semantic content negotiation.

REFERENCES

- [1] Mike Amundsen. 2017. *RESTful Web Clients: Enabling Reuse Through Hypermedia*. O’Reilly Media, Inc.
- [2] Nico Abfalq, Helen Schneider, and Tobias Käfer. 2021. Integrated Planning and Execution on Read-Write Linked Data. In *Proceedings of the All the Agents Challenge (ATAC) at the 20th International Semantic Web Conference (ISWC)*. 25–31.
- [3] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. 2007. *Developing Multi-Agent Systems with JADE*. John Wiley & Sons.
- [4] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons.
- [5] Rodney Allen Brooks. 1999. *Cambrian intelligence: The Early History of the New AI*. MIT press.
- [6] Victor Charpenay, Sebastian Käbisch, and Harald Kosch. 2016. Introducing Thing Descriptions and Interactions: An Ontology for the Web of Things. In *Joint Proceedings of the 3rd Stream Reasoning (SR) and the 1st Semantic Web Technologies for the Internet of Things (SWIT) workshops co-located with 15th International Semantic Web Conference (ISWC)*. 55–66.
- [7] Anthony Chemero and Michael Turvey. 2007. Complexity, Hypersets, and the Ecological Perspective on Perception-Action. *Biological Theory* 2, 1 (2007), 23–36.
- [8] Andrei Ciortea, Olivier Boissier, and Alessandro Ricci. 2018. Engineering World-Wide Multi-Agent Systems with Hypermedia. In *Weyns, D., Mascardi, V., Ricci, A. (eds) Engineering Multi-Agent Systems (EMAS) (Lecture Notes in Computer Science)*. 285–301.
- [9] Roy T. Fielding. 2000. *Architectural styles and the design of network-based software architectures*. Ph.D. Dissertation. University of California, Irvine.
- [10] Roy T. Fielding. 2008. REST APIs Must Be Hypertext-driven. Retrieved October 28, 2022 from <https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>
- [11] Richard E. Fikes and Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2, 3-4 (1971), 189–208.
- [12] Michael P. Georgeff and Amy L. Lansky. 1987. Reactive Reasoning and Planning. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI)*, Vol. 87. 677–682.
- [13] Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated Planning and Acting*. Cambridge University Press.
- [14] James J. Gibson. 2014. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology press.
- [15] Rex Hartson. 2003. Cognitive, Physical, Sensory, and Functional Affordances in Interaction design. *Behaviour & Information Technology* 22, 5 (2003), 315–338.
- [16] Tobias Käfer and Andreas Harth. 2018. Rule-based Programming of User Agents for Linked Data. In *Workshop on Linked Data on the Web (LDOW) co-located with The Web Conference 2018 (WWW)*.
- [17] Matthias Kovatsch, Michael McCool, Sebastian Käbisch, Victor Charpenay, and Takuki Kamiya. 2020. *Web of Things (WoT) Thing Description*. W3C Recommendation. W3C. <https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/>.
- [18] Yannis Labrou and Tim Finin. 1997. *A Proposal for a New KQML Specification*. Technical Report. Technical Report Technical Report TR-CS-97-03, University of Maryland.
- [19] Markus Lanthaler and Christian Gütl. 2013. Hydra: A Vocabulary for Hypermedia-driven Web apis. In *Proceedings of the 8th Workshop on Linked Data on the Web (LDOW) co-located with The Web Conference 2013 (WWW)*.
- [20] Markus Lanthaler, David Wood, and Richard Cyganiak. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
- [21] Jérémy Lemée, Danaï Vachtsevanou, Simon Mayer, and Andrei Ciortea. 2022. Signifiers for Affordance-driven Multi-Agent Systems. Paper presented at the International Workshop on Engineering Multi-Agent Systems (EMAS) at the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS).
- [22] Drew McDermott and Dejing Dou. 2002. Representing Disjunction and Quantifiers in RDF. In *Horrocks, L., Hendler, J. (eds) The Semantic Web (ISWC) 2002 (Lecture Notes in Computer Science)*. Springer, 250–263.
- [23] Mehdi Medjaoui, Erik Wilde, Ronnie Mitra, and Mike Amundsen. 2021. *Continuous API management*. " O’Reilly Media, Inc."
- [24] Don Norman. 2013. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books.
- [25] Donald A Norman. 2008. The Way I See IT Signifiers, Not Affordances. *Interactions* (2008).
- [26] Anand S. Rao, Michael P. Georgeff, et al. 1995. BDI Agents: From Theory to Practice.. In *Proceedings of the 1st International Conference on Multi-Agent Systems*, Vol. 95. 312–319.
- [27] Alessandro Ricci, Michele Pionti, and Mirko Viroli. 2011. Environment Programming in Multi-Agent Systems: An Artifact-based Perspective. *Autonomous Agents and Multi-Agent Systems* 23, 2 (2011), 158–192.
- [28] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. 2006. CArtaGO: A Framework for Prototyping Artifact-based Environments in MAS. In *Weyns, D., Parunak, H.V.D., Michel, F. (eds) Environments for Multi-Agent Systems III (E4MAS) 2006 (Lecture Notes in Computer Science)*. Springer, 67–86.
- [29] Yousouf Taghzouti, Danaï Vachtsevanou, Simon Mayer, and Andrei Ciortea. 2022. A Step Toward Semantic Content Negotiation. In *23rd International Conference on Knowledge Engineering and Knowledge Management (EKAW)*.
- [30] Farshid Tavakolizadeh, Andrea Cimmino, Michael McCool, and Kunihiko Toumura. 2023. *Web of Things (WoT) Discovery*. Candidate Recommendation. W3C. <https://www.w3.org/TR/2023/CR-wot-discovery-20230119/>.
- [31] Ruben Verborgh, Vincent Haerinck, Thomas Steiner, Davy Van Deursen, Sofie Van Hoeck, Jos De Roo, Rik Van de Walle, and Joaquim Gaborro. 2012. Functional Composition of Sensor Web APIs. In *Proceedings of the 5th International Conference on Semantic Sensor Networks (SSN)*. 65–80.
- [32] Gerhard Weiss. 2000. *Multiagent systems, Second Edition*. MIT Press.