Signifiers for Affordance-driven Multi-Agent Systems

Jérémy Lemée, Danai Vachtsevanou, Simon Mayer, and Andrei Ciortea

Institute of Computer Science, University of St. Gallen {jeremy.lemee, danai.vachtsevanou, simon.mayer, andrei.ciortea}@unisg.ch
https://ics.unisg.ch/chair-interactions-mayer/

Abstract. The ecological psychologist James J. Gibson defined the notion of affordances to refer to what action possibilities environments offer to animals. In this paper, we show how (artificial) agents can discover and use affordances in a Multi-Agent System (MAS) environment to achieve their goals. To indicate to agents what affordances are present in their environment and whether it is likely that these may help the agents to achieve their objectives, the environment may expose signifiers while taking into account the current situation of the environment and of the agent. On this basis, we define a Signifier Exposure Mechanism that is used by the environment to compute which signifiers should be exposed to agents in order to permit agents to only perceive signifiers that are likely to be relevant to them, and thereby increase their efficiency. If this is successful, agents can interact with partially observable environments more efficiently because the signifiers indicate the affordances they can use towards which purposes. Signifiers thereby facilitate the exploration and the exploitation of MAS environments. An implementation of signifiers and of a Signifier Exposure Mechanism is presented within the context of a Hypermedia Multi-Agent System and the utility and efficiency of this model is presented through the development of a scenario.

Keywords: Affordance · Signifier · Hypermedia · Multi-Agent Systems · Signifier Exposure Mechanism

1 Introduction

The field of Multi-Agent Systems (MAS) considers systems of agents acting within a shared environment to accomplish their goals [22]. In order to achieve their goals, the agents need to explore the environment to discover which actions could be undertaken. This exploration amounts to the discovery of *affordances*, a notion introduced by Gibson in the field of ecological psychology [7] to indicate what the environment affords an animal to do — and that we reuse here to indicate what a MAS environment affords to agents. The notion of affordance was also reused by designer Don Norman to indicate what a human-made thing affords to its users [14]: Don Norman studied how human-made things provide cues to indicate to their users how they ought to be used. Don Norman called such cues *signifiers*. Affordances and signifiers can be applied to the field of MAS. Indeed, a MAS environment offers affordances to agents, and signifiers can be integrated into the environment to enable agents to discover and interpret which affordances they can exploit.

This article shows how signifiers can be introduced as a first-class abstraction within a MAS environment so that agents can discover and use these signifiers to achieve their goals. A Signifier Exposure Mechanism (SEM) is introduced to manage the amount and content of signifiers that are perceived by agents. Agents can discover signifiers that are relevant to them by indicating their current states as well as their purposes. A special application to the case of hypermedia environments is made.

The introduction of signifiers into the environment can enable the use of plans in a manner suited to dynamic and only partially observable environments.

In Section 2, we discuss the ecological aspects of agent-environment interactions in systems composed of human and artificial agents that share a common environment, including MAS. In Section 3, our model for affordance-driven interaction in MAS is presented, and in Section 4 this model is implemented and evaluated. The model is then discussed in Section 5 with a conclusion in Section 6.

2 Agent-Environment Interaction in Socio-Technical Systems

J.J. Gibson defined *affordances* as the possibilities for behavior which the environment offers to animals [7]. The conception of the term enabled studying how animals control their behavior by perceiving and exploiting affordances towards achieving their goals. Additionally, affordance theory inspired many applied fields which aim to design physical or virtual environments for human or artificial agents, including the fields of Human-Computer Interaction (HCI) and MAS.

In this section, we present an overview of the related work that examines how autonomous agents perceive and exploit interaction possibilities offered by their environment within socio-technical systems. We examine approaches and methods that derive from both 1) HCI, where concepts from affordance theory were popularized and appropriated for computational systems, and 2) MAS, where such concepts have been also examined for enabling affordance-driven interaction in MAS.

In Section 2.1, we examine how interaction possibilities are commonly defined and conveyed by the environment to autonomous (human or artificial) agents. In Section 2.2, we examine how autonomous agents exploit perceived interaction possibilities towards interacting with their environment and achieving their design objectives.

2.1 Conveying Affordances in Socio-Technical Systems

The methods and mechanisms of conveying affordances concern the way affordances are defined and represented as well as communicated to human and artificial agents. How affordances are conveyed significantly affects how effectively agents discover and interpret interaction possibilities in their environments, thus the topic has been explored in both HCI and MAS.

Conveying Affordances in Human-Computer Interaction Don Norman popularised the concept of affordance in HCI when he formulated the fundamental principles of human agents' interactions within physical and virtual environments [14]. The proposed principles were constructed around the definition of affordance as "a relationship between the properties of an object and the capabilities of the agent that determine just how the object could possibly be used". Norman was particularly interested in designing environments that convey perceptual information about affordances, i.e. designing *perceived* affordances [5]. The design of perceived affordances facilitates the design of "everyday things" for which a human agent can easily infer what is afforded to her/him. To this end, Norman introduced the term *signifier* to denote *"any perceivable cue (deliberate or accidental) that can be interpreted meaningfully to reveal information about affordances"* [15, 14].

By conveying affordances through signifiers, affordances become discoverable and interpretable – two important features in designing products that can be exploited easily and intuitively by (human) agents [14]. In his latest work, Norman shifts completely the designers' attention from affordances to signifiers since their major design goal is to *convey* affordances through appropriate means (i.e., through signifiers) that reveal what the environment is for, and what the alternative possible actions within the environment are [15]. Additionally, introducing signifiers on top of affordances decouples the design of perceptual information from a specific affordance formalization (although Norman views affordances as relationships).

Alongside the physical world, affordances and signifiers are also prominent within another large-scale and affordance-rich environment: the hypermedia environment of the World Wide Web. Roy Fielding defines hypermedia by "the presence of application control information embedded within, or as a layer above, the presentation of information" [4]. He, further, notes that then information becomes "the affordance through which the user (or automaton) obtains choices and selects actions"[1] in the Web environment. These choices are provisioned by origin servers or intermediaries, which is captured in the "Hypermedia as the Engine of Application State" (HATEOAS) principle within the REST Uniform Interface constraint [4], i.e. a hypermedia control is an application control that is directly visible to the user. For example, the HTML format allows the definition of text that is clickable such that clicking on the text redirects the user to another Web page. The use of hypermedia controls gives a human user not only

[1] https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven

an affordance to visit another Web page but also a visual hint of the presence of this affordance, i.e. a signifier, since browsers usually color the text or render additional media that provide the hypermedia control in order to make it stand out from the rest of the document.

The presence of hypermedia controls on HTML pages allows human agents to explore the Web environment starting from a single entry point (such as a bookmark or the Web page of a search engine). This property can be replicated in all hypermedia environments in order to let any user (human or artificial) discover the affordances that are available to them as they dynamically explore the environment. Specifically, for enabling *artificial* agents to explore and interact in a manner within the Web environment similar to how humans do, hypermedia controls along with useful information about them should be presented within machine-readable documents. This was introduced as an envisioned contribution of the Semantic Web [1], a research area that has provided key concepts, methods and tools for knowledge representation and reasoning, unlocking new potentials for MAS composed of artificial agents. Exploiting Semantic Web technologies, the W3C Web of Things (WoT) initiative [2] standardized a formal model and a common machine-understandable representation for interactions that extend to the physical environment. As part of the W3C WoT Thing Description (TD)[3], so-called *Interaction Affordances* integrate hypermedia controls with higher-level semantics, enabling artificial agents to interpret and exploit affordances of devices that would have otherwise remained "hidden", i.e. where perceptual information about affordances would not have been available [5] and therefore would have needed to be provisioned through other means, such as API descriptions or other types of manuals. W3C TD Interaction Affordances thus become the *signifiers* that convey to artificial agents the types and the exploitation means of perceived affordances. As a result, artificial agents can discover W3C TDs as any other Web document and thereby dynamically explore the hypermedia environment.

Conveying Affordances in Multi-Agent Systems Affordances have also been utilized for artificial agents that are not situated in hypermedia environments, and especially in agent-based simulations of human behavior (e.g., [9, 19, 11, 17]). For example, an approach that views affordances as relationships in the context of affordance-based simulated agent environments comes from the work of Klügl and Timpf [11]. The authors model each affordance offered by a potential interacting partner (i.e., an agent or an object in the environment) as a relationship between an agent, the interacting partner, an activity (i.e., a target behavior), and a priority value. The priority value provides a direct way of affecting how the affordance is conveyed and perceived by an agent. The model specifies that the priority depends on the *level of complementarity* between the agent and the interacting partner. Although the authors do not analyze how this complementarity is represented, we could imagine that prioritisation in percep-

[2] https://www.w3.org/WoT/
[3] https://www.w3.org/TR/wot-thing-description/

tion could rely on the popularity of an affordance estimated by the designer of the environment. This would require that the designer adopts a user-centred perspective for designing perceptual information about affordances by considering specific types of agents that will inhabit the environment.

In [17], Papasimeon uses the notion of "annotation", which is a label put in the environment, and identifies an affordance as a special class of annotation. Among annotations, affordances are characterized by being:

- Action-Oriented: an affordance provides actions.
- Meaningful: An affordance is related to the goals of the agent.
- Relational: An affordance is a relation between an agent and an environment.
- Intentional: The agent's intentions influence its perception of affordances.
- Directly perceivable: Agents can directly perceive the available actions.

Papasimeon's thesis was published in 2010, that is around the time period when Norman remarked that an affordance is present whether perceptual information about it is present or not. Therefore, perceptual information is not an affordance but a signifier — a cue for conveying an affordance. This criticism can be applied to Papasimeon's thesis which defines affordance as a subtype of annotation, although *signifier* is rather a subtype of annotation, but affordance is not. This results in an intermingling of the design objectives of signifiers and affordances. On the other hand, one could separate the concerns between the two concepts, and concequently, decouple the relational and objective aspects of affordance-driven interaction: Affordances remain relational, while signifiers can be designed on top carrying objective semantics.

On another approach, Joo et al. [9] utilizes an affordance-based interaction formalisation [10] that considers affordances as properties of the environment. By bringing affordances to the computational space, the authors aim to simulate more realistically the availability and exploitation of interaction possibilities in dynamic environments. Their use case concerns the problem of wayfinding for agents during an emergency evacuation scenario. Again, information about affordances is conveyed *by the environment* – in this case, the building that is being evacuated. Specifically, based on its current state, the environment updates dynamically the available information about its affordances. For example, if at a certain point in time, some space in the simulated physical environment gets occupied by an object, then the information about the affordance *occupy-ability* of the space is updated in the environment. An agent perceiving the environment will not only perceive the space, but also a numeric value regarding the current occupy-ability of the space, indicating whether occupying the space is possible or not. This approach transfers computational load from the agent to the environment, since the agent does not need to keep information about the possible system states, or continuously reason about the availability of interaction possibilities. In turn, the agent is responsible for observing its environment every time it is about to proceed with an interaction. Again, it needs to be noted that the conveyed numerical values are functioning more like signifiers rather than affordances, since they are modelled and calculated based on affordance instances.

In his 2001 article [19], Raubal studies the problem of wayfinding for agents on a more abstract level. The paper discusses that agents are able to follow a path to reach their destination in an environment by discovering and interpreting signs positioned in the environment. By following these signs that act as knowledge in the world, an agent is able to reach its destination without constructing a mental model of a potentially dynamic or nondeterministic environment. Because these signs are perceivable and communicate appropriate behavior to an agent, they are signifiers according to Don Norman's definition. These signifiers can be added to the environment. If these signifiers are meaningful (i.e. related to the goals of the agent according to Papasimeon), then they can be used to indicate personalized information to an agent in order to let that agent achieve its desired goal, which may be different from the goal of another agent.

2.2 Exploiting Affordances in Goal-driven Behavior

Any proactive autonomous agent exhibits goal-driven behavior, i.e. it takes the initiative to exploit interaction possibilities for satisfying its goals [22]. Goal-driven behavior entails that an agent has a goal deliberation process for computing which goals to be pursued, suspended, or abandoned at a given system state, as well as a decision-making process for determining which course of action to take (i.e., which affordances to exploit) in order to drive the system state towards its preferences, for instance, towards a preferred state of the environment. Affordance-driven behavior has been explored in the context of goal-driven behavior both in HCI and MAS.

Exploiting Affordances in Human-Computer Interaction Goal-driven behavior in affordance-based systems differs based on how affordances are conveyed, i.e. what type of signifiers agents perceive in their environment. Norman specifies that facilitating human agents' goal-driven behavior requires minimizing the "gulf of execution"[14]. This indicates how well affordances of the environment address the agent's intentions, and how appropriate signifiers are for facilitating the discoverability and interpretability of said affordances. Specifically, once an agent deliberates over its goals and has the intention to act, it plans the course of action required to achieve the goals. During this step, if signifiers indicate what an affordance *is for* or *how relevant* an affordance is estimated to be for the agent, then they can provide a useful means during planning. After specifying the course of action based on the currently exploitable and relevant affordances, the agent executes the desired actions. During this step, if signifiers indicate *how to* exploit affordances, then they also provide a useful means during execution.

On the Web environment, the way hypermedia is presented to humans (e.g., with a specific text color, label, or icon) expresses what the underlying affordance is for. This enables human agents to plan their next action. Through hypermedia, executing a required action is also very intuitive since human agents exploit affordances in a uniform manner, simply by interacting with the perceived affordances (i.e., the presented text or other media), while the browser is responsible for any lower-level details of the interaction (i.e., for using the hypermedia controls). After exploiting an affordance, the system state progresses, and the human agent encounters a new set of hypermedia, i.e., a new set of valid state transitions and related perceptual information. As a result, human agents are not required to plan ahead every step towards achieving their goals, but they rather determine their next action after exploiting an affordance. Thus, interaction becomes affordance-driven, and, in this case, hypermedia-driven due to HATEOAS. Based on Norman, human agents' goals and intentions on the Web are mainly *opportunistic*, resulting in less mental effort [14].

For artificial agents on the Web, W3C WoT TD Interaction Affordances provide the ground for exploiting affordances of physical devices in a similar opportunistic manner. Currently, multi-step interactions can be supported through a proposed extension to TD for modelling and representing more complex behaviors of devices in the form of *paths* [12]. Paths describe meaningful valid sequences of interaction possibilities with respect to the state of a device. This approach further decouples agents from entities providing interaction possibilities and alleviates the load of pre-defining knowledge about the internal states and transitions that drive the operation of devices, planning, or observing the environment before every interaction.

As an alternative, RESTdesc [21] enables planning over information about interaction possibilities provided through hypermedia. RESTdesc defines a format for specifying hypermedia controls with respect to preconditions and postconditions of exploiting them. RESTdesc descriptions enable artificial agents to interpret how to exploit interaction possibilities, as well as to reason about when an interaction possibility is exploitable and what are the consequences of such an exploitation. Both TD paths and RESTdesc descriptions facilitate the goal-driven behavior of artificial agents, since the former impose less computational load on agents by providing cohesive pre-defined sequences of interactions, while the latter induce more interaction flexibility via reasoning for HATEOAS (at the cost of computational resources [13]).

Exploiting Affordances in Multi-Agent Systems Different methods for exploiting affordances in goal-driven interactions are also encountered in related work in MAS. In [9], an agent, initially, computes a "rough" plan based on prior knowledge that is then resolved into a detailed plan based on dynamically perceived affordances of the environment. Prior knowledge includes application-specific and scenario-specific information that help the agent to compute a plan that will be later refined. For example, an agent already knows that a certain sub-goal state `A` brings the state of the application closer to the desired final-goal state than another sub-goal state `B` does. Thus, the agent is more likely to compute a plan that includes sub-goal `A` rather than sub-goal `B`. This assumption is appropriate for physical environments whose structure rarely changes and for environments that are familiar to the agents occupying them. On the other hand, in case the environment exhibits greater dynamics in its structure (e.g., in the case of a virtual environment), the integrity of such static prior knowledge cannot be assumed. We argue that in open and dynamic environments which continuously evolve and expand to include new or updated states, an agent should a) update frequently its goal-independent knowledge about the structure of the environment (e.g., by dynamically acquiring a map of the environment) or b) rely less on an internal planning module that depends on global information about the environment.

In [8], Joo et al. extend their previous work to MAS composed of Belief-Desire-Intention (BDI) agents. BDI theory is used to model the information (beliefs) and the motivational (desires) and deliberative (intentions) states of agents [18], and affordance theory is used to model the affordance-effectivity pairs that enable agents to make decisions to take action. Here, planning remains relatively independent of the affordances that have been perceived, and affordance availability is mainly computed to reject or refine a plan if needed, and then to execute the finalized plan. Specifically, a BDI reasoning processor is responsible for generating an agent's intentions, i.e. the goals which the agent is committed to achieve. Planning is performed based on the agent's beliefs and results in the optimal route of actions for satisfying an intention. After this first planning phase, affordance-effectivity pairs [10] are used to screen out those actions that are currently not possible (triggering a re-planning phase if required). After the computation of a realizable plan, each action of the plan is executed when the action becomes possible based on the set of affordance-effectivity pairs.

While Joo et al. [8] mainly separate the concerns between the BDI-based and affordance-based system components, Papasimeon designs an affordance-based BDI reasoning processor. Since in [17] affordances are considered to be meaningful and intentional (as discussed in Section 2.1), agents' perceived affordances already provide useful input for reasoning and planning towards achieving related intentions. Thus, an agent may adopt, i.e., decide to exploit, any of the perceived affordances, acknowledging that exploitation is appropriate for its goal-driven behavior. At this point, once an affordance has been adopted, a new intention is generated that commits the agent to execute the affordance-related action. As a final step to the agent reasoning model, the agent achieves its affordance-related intentions by executing the corresponding actions.

Hypermedia Multi-Agent Systems Hypermedia Multi-Agent Systems fall into a specific class of MAS where autonomous agents are situated in a distributed hypermedia environment (e.g., see [3, 16]). A hypermedia environment is an environment that can be explored and exploited through hypermedia affordances. A Hypermedia MAS is characterized by three principles [3]:

- Uniform resource space: All entities in a Hypermedia MAS and relations among them are represented in the hypermedia environment in a uniform, resource-oriented manner.
- Single entry point: Given a single entry point into the environment of a Hypermedia MAS, an agent should be able to discover the knowledge required to participate in the system by navigating the hypermedia.

– Observability: In a Hypermedia MAS, any resource in the hypermedia environment that could be of interest to agents should be observable.

These principles enable and complement HATEOAS for artificial agents. As a result, the hypermedia environment provides to an agent the affordances to explore the environment starting from one entry point, and any affordance available and of interest to an agent can be discovered by that agent, and exploited through the provided hypermedia controls. Thus, Hypermedia MAS seem to offer a viable infrastructure for conveying signifiers in a manner that enables affordance-driven interaction in MAS.

3 Affordance-driven Interaction in Multi-Agent Systems

Agents need to use affordances to achieve their goals. In order to let agents perceive affordances, a model for signifiers indicating the affordances present in the environment and their potential uses is created in Section 3.1. A Signifier Exposure Mechanism is developed in Section 3.2 to let agents only perceive the signifiers that are relevant to them. In Section 3.3, an explanation of how affordances can be integrated into the plans of the agents is given.

3.1 Modelling Affordances and Signifiers

Definition 1 *An* affordance *is a relation between an agent and its environment indicating what an agent can or cannot do within the environment and what the agent can achieve by using the affordance.*

An affordance is therefore available to an agent under certain conditions and using the affordance changes the conditions of the environment. The notion of *state* is introduced to model these conditions.

Definition 2 *A state is a partial description of the environment that contains a set of statements that are true in the environment and a set of statements that are false in the environment.*

A state `S1` implies a state `S2` if the statements that are true in `S1` are also true in `S2`, and if the statements that are false in `S1` are also false in `S2`. It is now possible to define the components of the description of an affordance.

Preconditions and Postconditions The preconditions and postconditions of affordances carry the same semantics as these concepts do for actions (or operators) in classical planning [6]: They denote the conditions for applying a related possible action (i.e., exploiting an affordance), and respectively, the conditions that hold after executing the possible action (after exploiting the affordance). The difference is that such conditions remain attached to affordances, and decoupled from specific and grounded actions which are only encountered and executed (become possible) at run-time.

Objective An objective is a state that describes a goal that an agent can achieve through using the affordance. Contrary to the postcondition, the objective is not necessarily immediately reached when the affordance is used, but using the affordance will bring the agent closer to achieving this objective. If an agent has a given objective, this agent can compare this to the objectives associated to the affordance by signifiers. If there is a match between the agent's objective and the affordance's objectives, then the agent learns that using the affordance will bring it closer to achieving its objective. Therefore, an agent can directly follow affordances that match its objectives in order to reach the objective, which allow the agent to reach its goal without performing any planning, which would be needed if the agent only knew the precondition and the postcondition of the affordances. An affordance can have many objectives and the description may not be exhaustive.

Action An action indicates how an agent can use the affordance when the precondition is satisfied. The actions available to an agent are relative to the environment. In a hypermedia environment, an action is a request using a protocol like HTTP[4] or CoAP[5].

Definition 3 *A signifier provides information explaining how to discover and exploit affordances.*

The signifier contains the description of an affordance but it may also contain other metadata, as described in the following:

- *Expiration Date*: The expiration date indicates until when a signifier is considered valid. After this date, the description of the affordance provided by the signifier may no longer hold. If the expiration date is not present, the agent using the signifier should assume that the signifier is valid and will remain valid in the foreseeable future.
- *Location*: The location indicates the position of the signifier within the environment. This location can be logical (e.g workspace, artifact in the the Agents & Artifacts meta-model [20]) or physical (e.g geographical coordinates).
- *Creator*: The creator of the signifier may be the designer of the environment or an agent. The knowledge of the creator can determine the level of trust the agent has in the signifier.
- *Salience*: The salience is an objective measure of the expected relevance of the signifier defined by the creator of the signifier. A high salience indicates that, a priori, the signifier will be relevant to an agent that is a stereotypical user, according to the creator of the signifier, while a low salience indicates that the signifier will only be useful to a limited class of agents.

[4] https://datatracker.ietf.org/doc/html/rfc2616
[5] https://datatracker.ietf.org/doc/html/rfc7252

3.2 Perception of Affordances through the Signifier Exposure Mechanism

Agents need to perceive signifiers in order to gain information to achieve their goals. Such signifiers are relevant signifiers. Depending on the situation, all relevant signifiers may be necessary to let an agent achieve its goal or only one such signifier is necessary. Because of that, letting the agent perceive all signifiers might be highly inefficient especially if the agent has limited processing ability or memory. In order to solve this problem, a Signifier Exposure Mechanism can be added to the environment.

Definition 4 *A Signifier Exposure Mechanism (SEM) is a mechanism by which an agent perceives signifiers from its environment.*

The Signifier Exposure Mechanism is programmed by the designer of the environment and there is no constraint about which signifiers it should let the agent perceive. However, in order to make the Signifier Exposure Mechanism helpful to the agent, the Signifier Exposure Mechanism can filter all the available signifiers and send only the ones it estimates to be relevant. Such a filtering can be done with the information that the Signifier Exposure Mechanism possesses about the content of the signifiers and the state of the environment. For example, if a signifier is given a location and the agent is present at that location then the signifier is made visible to the agent, if the agent is not present at that location then the signifier is not made visible to the agent. The salience can also be used by the Signifier Exposure Mechanism. Indeed, the salience represents an a priori level of relevance. Since a salient signifier is more likely relevant than one that is not, salient signifiers should be given higher priority.

However, these approaches do not take into account the agent's preferences and abilities. Since the agent is the ultimate arbitrator to decide whether a signifier is relevant or not, it is more efficient to take into account the agent's preferences and abilities when doing the filtering of signifiers. However, the Signifier Exposure Mechanism does not have direct access to them because they are only internal to the agent. However, an agent can create an agent profile to indicate its preferences and abilities (including the agent's knowledge, such as knowledge of a password to access a given resource). If the Signifier Exposure Mechanism is given access to this profile, it can use its content to improve the filtering of signifiers. For example, if the agent's abilities prevents it from using an affordance or if an affordance's objectives do not match the agent's objectives, the signifier describing that affordance may remain hidden to the agent. However, the agent is the ultimate decider of whether a signifier is relevant or not. If a signifier perceived by the agent is deemed irrelevant, the agent may discard it and if an agent does not receive any signifier it finds relevant, it can update its profile in order to find more relevant signifiers.

3.3 Usage of Affordances in Agent Plans

In classical planning, a plan contains a sequence of actions where each action is characterized by a precondition and a postcondition. In a dynamic environment, this condition implies that agents need to replan in case any precondition do not hold during execution, for instance, due to actions executed by other agents during planning. In such cases, affordance-driven interaction could provide an alternative approach for handling the environment dynamics. Specifically, an agent does not need to plan ahead, but it can opportunistically exploit affordances based on the currently perceived signifiers: using a Signifier Exposure Mechanism, the agent only perceives signifiers that are currently exploitable by it as well as relevant to it, thus optimizing its perception towards its effective goal-driven behavior.

Additionally, we can consider the case where an agent is already familiar with a set of pre-defined signifiers. In this case, planning could be achieved based on the preconditions and postconditions described in the signifier, while staying decoupled from specific actions. As a result, an agent could be able to plan not based on actions that potentially are currently not exploitable, but, instead, based on preconditions and postconditions of affordance types. Eventually, the actions in these plans are not the actions of the environment but are actions that indicate what the agent should look for in the environment. For example, such an action would be to look for an affordance in the environment, parameterized with the affordance type of interest. If the agent finds an affordance matching the given type, then the agent knows it can use an action provided by this affordance towards achieving its goals.

4 Implementation and Experience

The model that we develop for signifiers was implemented with a library to process signifiers, a creation of hypermedia artifacts with signifiers on the platform Yggdrasil[6] [3], and the creation of a JaCaMo project [2] to create agents able to use hypermedia artifacts with signifiers on Yggdrasil in Section 4.1. We tested our implementation with a maze scenario developped on Yggdrasil and the JaCaMo project in Section 4.2.

4.1 Implementation

Three ontologies are developped: an ontology to model signifiers [7], an ontology to model agent profiles [8], and an ontology to model hypermedia actions [9]. A Java library is created, based on these ontologies, in order to create and processed signifiers based on that ontology. This library is based on the library RDF4J that is used to process RDF in Java. It therefore enables the serialization and deserialization of signifiers.

[6] https://github.com/Interactions-HSG/yggdrasil
[7] https://w3id.org/interactions/ontologies/signifiers/v1
[8] https://w3id.org/interactions/ontologies/profile/v1
[9] https://w3id.org/interactions/ontologies/hypermedia/v1

Yggdrasil is a framework coded in Java, based on the framework `Vert.x`[10], and used to create hypermedia environments that conform to the Agents & Artifacts meta-model. Within this platform, hypermedia artifacts are used to create artifacts whose operations are availble to agents on the Web. A special class of hypermedia artifacts with signifiers is created with an operation to let an agent retrieve the URIs of the signifiers of the artifacts that are visible to the agent. A hypermedia artifact with signifiers is able to read the profile of an agent from an agent profile artifact in order to provide a more efficient filtering of signifiers.

Agents are developed to use the hypermedia artifacts developed on Yggrasil. The project jacamo-signifiers-hypermedia [11], developed on the JaCaMo platform using the signifier-java library, is used to program the agents as well as the artifacts that are used to interact with Yggrasil. The main artifact created to interact with Yggrasil is the HTTPArtifact that allows an agent to create any HTTP request and can also create an HTTP request from a hypermedia action.

4.2 Scenario: Maze

A maze-crossing scenario is developed as a toy problem to show how an agent can use signifiers and what benefits signifiers bring to the agent compared to a solution that does not use signifiers.

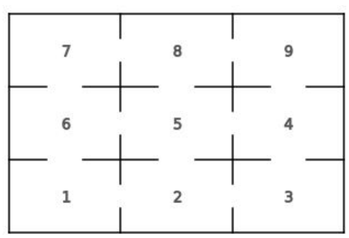

Fig. 1: Representation of the maze used in the scenario.

In this scenario, the agent is in a maze (Figure 1) represented as a hypermedia artifact on Yggdrasil. The agent starts the maze in room 1 and decides to reach another room. The signifiers present in the maze, as well as the Signifier Exposure Mechanism can help to agent to cross the maze efficiently.

Listing 1.1: Signifier representation for a movement affordance

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix sig: <https://w3id.org/interactions/ontologies/signifiers/v1#>.
@prefix hyp:<https://w3id.org/interactions/ontologies/hypermedia/v1#>.
@prefix ex: <http://example.org>.
```

[10] https://vertx.io/

[11] https://github.com/Interactions-HSG/jacamo-signifiers-hypermedia

```
_:signifier12 a sig:Signifier;
sig:hasAffordance _:affordance12 .

_:affordance12 a sig:Affordance;
sig:hasPrecondition _:room1Id;
sig:hasPostcondition _:room2Id;
sig:hasObjective _:room9Id;
sig:hasAction _:movement1 .

_:room1Id a sig:State;
sig:hasStatement _:room1S .

_:room1S rdf:subject sig:thisAgent;
rdf:predicate ex:isIn;
rdf:object ex:room1.

_:room2Id a sig:State;
sig:hasStatement _:room2S .

_:room2S rdf:subject sig:thisAgent;
rdf:predicate ex:isIn;
rdf:object ex:room2.

_:movement1 a hyp:HypermediaAction;
hyp:hasUrl "http://localhost:8080/environments/env1/workspaces/wksp1/artifacts/maze/move";
hyp:hasMethod "POST";
hyp:hasPayload "[1]".
```

The scenario consists of the following phases:

1. The agent registers to the maze artifact in room 1. The maze artifact adds the identifier of the agent and its position (room 1) to its database.
2. The agent creates a profile indicating that it wishes to go to room 9. The agent may also add an upper limit of 1 signifier received per request in order to reduce the number of signifiers to process.
3. The agent publishes this profile to the maze.
4. In each room starting from room 1, if the room is not the agent's destination, then the agent retrieves the visible signifier. Each retrieved signifier (such as the one in Listing 1.1) indicates an affordance that brings the agent closer to its destination. For example, in room 1, the signifiers shown are the one to go from room 1 to room 2 and the one to go from room 1 to room 6 (if the agent asks for a limit of 1 signifier, then only one of them is shown to the agent).

4.3 Evaluation

The scenario illustrates different benefits are brought by the introduction of signifiers:

- Each room contains the signifiers that indicate the available affordances to reach the adjacent room as well as information concerning the objectives that may be achieved by using these affordances. These signifiers can be added either by the creator of the artifact or by another agent. Since each signifier indicate how to bring the agent closer to its objective, the agent does not need to know the map of the maze in order to cross it while still being able to follow the shortest path its destination. Indeed, the path 1⇒2⇒3⇒4⇒9 followed by the agent using the signifiers is indeed the shortest path between room 1 and room 9 with 5 rooms visited. The path followed is the shortest because the path provided by the signifiers is the shortest. If the information provided by the signifiers is not complete or wrong, the agent would still be able to find out a path through exploration and reasoning but the path would not necessarily be the shortest. Indeed, an agent with no knowledge of the structure of the maze could follow the path 1⇒2⇒3⇒4⇒5⇒6⇒7⇒8⇒9 where all 9 rooms are visited, and the path could even be longer if an agent can visit one room many times. Since the agent can choose to use or not the information provided by the signifiers, the signifiers do not restrict the autonomy of the agent, but only acts as sources of information.

- The Signifier Exposure Mechanism reduces the computational cost of the agent. Indeed, after the agent presents its profile to the Signifier Exposure Mechanism, the Signifier Exposure Mechanism only shows to the agent relevant signifiers, that is signifiers that brings to agent closer to its destination. Therefore the agent does not need to process signifiers that will not lead it to its destination.
- If the agent decides to add a limit of one signifier received per request, then the agent receives less signifiers, which is therefore a gain of memory for the agent. This gain of memory can be useful in environments with many signifiers and when the agent has limited memory. However, it requires the agent to explicitly indicate this limit into its profile and the agent may not receive some relevant signifiers. However, in the scenario, only one relevant signifier is needed at each step and therefore the agent can use a limit of 1 signifier without missing any necessary information.

5 Discussion

Human beings are able to use signifiers present in the environments in order to achieve their objectives with little to no a priori knowledge of the environment. For example, in a building, a human being does not need to know the map of the building how to reach the exit if exit signs are present in the building because the human beings just follows the signs. This property can be applied to environments created to be used by software agents with a special application to hypermedia environments.

The ontology developed for signifiers can be read by any agent able to process RDF documents. With this ontology, agents can understand and use any signifier provided they also know the ontology used to describe the actions in the environment. In our implementation, we define hypermedia actions that represent the HTTP requests an agent need to perform in order to use the affordances in the hypermedia environment.

The Signifier Exposure Mechanism allows a selective presentation of signifiers to the agent. Contrary to signs that are visible to all agents at the right location, such as road signs for drivers, the Signifier Exposure Mechanism also takes into account the agent's own preferences indicated by the agent profile in order to show the agent only the signifiers it needs to use. Therefore, the Signifier Exposure Mechanism can reduce the processing of signifiers by the agents to a minimum.

An agent can also possess a partial plan where the actual actions to perform are not indicated. Instead, the agent is given the type of the affordance to use, with the type being either a label or the objective of the affordance. If the agent perceives a signifier indicating an affordance with the proper type, the agent can use the action of the affordance. These partial plans can be used in dynamic environment where the agent does not know the actual affordances it will find but knows the types of these affordances. The knowledge of these partial plans reduces the need for planning because one single partial plan could be potentially used in many environments or in the same dynamic environment at different times.

6 Conclusions

In order to provide agents with the information about the affordances they can use to achieve their goals, signifiers are added to the environment. A model for signifiers, and an ontology based on this model, are created. A Java Library is programmed to read, create, and process signifiers based on this ontology. In order to let agents perceive only the signifiers that are relevant to them, the environment implements a Signifier Exposure Mechanism that indicate which signifiers can be perceived by the agents. The Signifier Exposure Mechanism can rely on an agent profile, created by the agent, in order to have a more efficient filtering of signifiers. In order to create a hypermedia environment with signifiers, the platform Yggdrasil is extended with hypermedia artifacts that contain signifiers and is given an implementation for the Signifier Exposure Mechanism. One scenario is implemented with agents created on the JaCaMo platform with artifacts to interact with the hypermedia artifacts containing signifiers on Yggdrasil. The scenario demonstrates the benefits of using signifiers and the Signifier Exposure Mechanism.

Future research can be done to integrate the signifiers created with the OWL ontology, that are machine-readable, with signifiers that are human-readable in order to facilitate the interactions between humans and software agents within a shared hypermedia environment.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation program under grant No. 957218 (*IntellIoT*) and from the Swiss National Science Foundation under grant No. 189474 (*HyperAgents*).

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american (2001)
2. Boissier, O., Bordini, R., Hübner, J., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. In: Science of Computer Programming (2013)
3. Ciortea, A., Boissier, O., Ricci, A.: Engineering world-wide multi-agent systems with hypermedia. In: EMAS 2018. pp. 285–301. Springer (2018)
4. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis (2000)
5. Gaver, W.W.: Technology affordances. In: Proc. of SIGCHI (1991)
6. Ghallab, M., Nau, D., Traverso, P.: Automated planning and acting. Cambridge University Press (2016)
7. Gibson, J.J.: The ecological approach to visual perception: classic edition. Psychology Press (2014)
8. Joo, J.: Perception and bdi reasoning based agent model for human behavior simulation in complex system. In: Hum. Comput. Interact. pp. 62–71. Springer (2013)
9. Joo, J., Kim, N., Wysk, R.A., Rothrock, L., Son, Y.J., Oh, Y.g., Lee, S.: Agent-based simulation of affordance-based human behaviors in emergency evacuation. Simulation Modelling Practice and Theory **32**, 99–115 (2013)
10. Kim, N., Joo, J., Rothrock, L., Wysk, R., Son, Y.J.: Human behavioral simulation using affordance-based agent model. In: Hum. Comput. Interact. pp. 368–377. Springer (2011)
11. Klügl, F., Timpf, S.: Towards more explicit interaction modelling in agent-based simulation using affordance schemata. In: German Conference on AI (Künstliche Intelligenz). pp. 324–337. Springer (2021)
12. Korkan, E., Kaebisch, S., Kovatsch, M., Steinhorst, S.: Safe interoperability for web of things devices and systems. In: Languages, Design Methods, and Tools for Electronic System Design. Springer (2020)
13. Kovatsch, M., Hassan, Y.N., Mayer, S.: Practical semantics for the internet of things: Physical states, device mashups, and open questions. In: IOT 2015. pp. 54–61. IEEE (2015)
14. Norman, D.: The design of everyday things: Revised and expanded edition. Basic books (2013)
15. Norman, D.A.: The way I see it signifiers, not affordances. Interactions (2008)
16. O'Neill, E., Lillis, D., O'Hare, G.M.P., Collier, R.W.: Delivering multi-agent microservices using cartago. In: Baroglio, C., Hubner, J.F., Winikoff, M. (eds.) Engineering Multi-Agent Systems. pp. 1–20. Springer International Publishing, Cham (2020)
17. Papasimeon, M.: Modelling agent-environment interaction in multi-agent simulations with affordances. Tech. rep. (2010)
18. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. In: Icmas. vol. 95, pp. 312–319 (1995)
19. Raubal, M.: Ontology and epistemology for agent-based wayfinding simulation. Int. J. Geogr. Inf. Syst. **15**(7), 653–665 (2001)
20. Ricci, A., Piunti, M., Viroli, M.: Environment programming in multi-agent systems: an artifact-based perspective. Auton. Agents Multi-Agent Syst. **23**(2), 158–192 (2011)
21. Verborgh, R., Haerinck, V., Steiner, T., Van Deursen, D., Van Hoecke, S., De Roo, J., Van de Walle, R., Gabarro, J.: Functional composition of sensor web apis. In: SSN. pp. 65–80 (2012)

22. Wooldridge, M.: Intelligent agents. Multiagent systems, Second Edition (2013)