Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-024-09938-6

Signifiers for conveying and exploiting affordances: from human-computer interaction to multi-agent systems

Jérémy Lemée[1] · Danai Vachtsevanou[1] · Simon Mayer[1] · Andrei Ciortea[1,2]

Accepted: 10 March 2024

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 exploit 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 information about affordances that are likely to be relevant to them, and thereby increase their interaction efficiency. If this is successful, agents can interact with partially observable environments more efficiently because the signifiers indicate the affordances they can exploit towards given purposes. Signifiers thereby facilitate the exploration and the exploitation of MAS environments. Implementations of signifiers and of the Signifier Exposure Mechanism are presented within the context of a Hypermedia Multi-Agent System, and the utility of this approach is presented through the development of a scenario.

Keywords Affordance · Signifier · Hypermedia · Multi-agent systems · Exposure mechanism

This is an extended version of the paper "Signifiers for Affordance-driven Multi-Agent Systems"[1] that was presented at the 10th International Workshop on Engineering Multi-Agent Systems (EMAS 2022).

✉ Jérémy Lemée
jeremy.lemee@unisg.ch

Danai Vachtsevanou
danai.vachtsevanou@unisg.ch

Simon Mayer
simon.mayer@unisg.ch

Andrei Ciortea
andrei.ciortea@unisg.ch

1 Institute of Computer Science, University of St. Gallen, Rosenbergstrasse 30, St. Gallen 9000, Canton of St. Gallen, Switzerland

2 WIMMICS, Inria, Université Côté d'Azur, CNRS, I3S, Sophia Antipolis, France

Published online: 17 April 2024

Mathematics Subject Classification (2010) 68T42

1 Introduction

The field of Multi-Agent Systems (MAS) considers systems of agents acting within a shared environment to accomplish their goals [2]. 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 James J. Gibson in the field of ecological psychology [3] to indicate what the environment affords to an animal— and that we reuse here to indicate what a MAS environment affords to agents. The notion of affordance was also reused by Don Norman to indicate what a human-made thing affords to its users [4]: Norman studied how human-made things provide cues to indicate to their users how they ought to be used. He called such cues *signifiers*. We propose to apply the concepts of affordances and signifiers to the field of MAS.

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. In [5], the authors formalize signifiers and a signifier management mechanism in Web-based MAS to facilitate agents' interaction in large-scale and open environments. In correspondence with this contribution, we provide a comprehensive literature review on domains relevant to socio-technical systems in the current article – this establishes the conceptual foundation that motivates the introduction of signifiers as a first-class abstraction in MAS. Our review encompasses the origins and adoption of signifiers and related concepts, focusing on their role in conveying and exploiting affordances in both Human-Computer Interaction and MAS, spanning physical and virtual environments. We present a model for representing information about interaction possibilities through signifiers in MAS, which addresses fundamental aspects of signifiers, and discuss the core features of dynamic signifier publication, exposure, and perception. The model complements the introduced concept of a Signifier Exposure Mechanism (SEM) that aims at managing the amount and content of signifiers that are exposed to agents based on the dynamic agent-environment complementarity. Specifically, this permits agents to be supplied with signifiers that are relevant to them through an SEM, which requires them to indicate their current situation (state and context) as well as their objectives. A special application to the case of hypermedia environments is presented as a prototype implementation that adheres to the approach and demonstrates the features of dynamic signifier exposure and publication. Our work motivates the introduction of signifiers into the environment, and the management of signifier exposure based on the run-time agent-environment complementarity, as an enabler for the exploitation of affordances in a manner suited to dynamic and only partially observable environments — comparatively, in a more affordance-driven manner.

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 environments. 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.

2 Agent-environment interaction in socio-technical systems

James J. Gibson defined *affordances* as the possibilities for behavior which the environment offers to animals [3]. 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 (human as well as artificial) autonomous agents perceive and exploit interaction possibilities offered by their environment within socio-technical systems. We examine approaches and methods that derive from both: HCI, where concepts from affordance theory were popularized and appropriated for computational systems; and MAS, where such concepts have been examined for enabling affordance-driven interaction.

In Section 2.1, we examine how interaction possibilities are commonly defined and conveyed by the environment to autonomous agents. In Section 2.2, we examine how autonomous agents exploit conveyed interaction possibilities, towards meaningfully interacting with the 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 environment, thus the topic has been explored in both HCI and MAS.

2.1.1 Conveying affordances in human-computer interaction

Don Norman popularised the concept of affordance in HCI when he formulated the fundamental principles of human interactions with physical and virtual environments [4]. 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. in designing *perceived* affordances [6]. 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* [4, 7].

By conveying information about affordances through signifiers, affordances become discoverable and interpretable – two important features when designing products that can be exploited easily and intuitively by (human) agents [4], ideally without requiring additional out-of-band information such as a user manual. In his latest work, Norman shifts completely the designers' attention from affordances to signifiers, arguing that a designer's major goal when designing an environment (and the "things" in it) 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 [7]. Additionally, introducing signifiers on top of affordances decouples the design of perceptual information from the mathematical formalisation of affordances (although Norman views affordances as relationships) – a process that still remains under investigation in the field of ecological psychology [1].

[1] We refer the interested reader to [8], where the authors present and review the major approaches in affordance formalisation.

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 as "the presence of application control information embedded within, or as a layer above, the presentation of information" [9]. He further notes that, as a result, information becomes "the affordance through which the user (or automaton) obtains choices and selects actions"[2] in the Web environment. These choices are provisioned by origin servers or intermediaries based on the Uniform Interface constraint of the Representational State Transfer architectural style [9]. The Uniform Interface constraint specifies that interaction possibilities on the Web are offered and exploited in a uniform hypermedia-driven manner, so that *Hypermedia is used as the Engine of Application State* (this is referred to as the "HATEOAS" principle). As a result, hypermedia controls become application controls which are both perceivable and exploitable by users. For example, the HTML format allows the definition of text that is clickable so that clicking on the text redirects the user to another Web page. The use of hypermedia controls gives a human user not only 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 within the Web environment in a manner 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 [10], a research area that has provided key concepts, methods and tools for knowledge representation and reasoning, unlocking new potentials for Web-based MAS composed of artificial agents. Taking a step further in exploiting Semantic Web technologies, the W3C Web of Things (WoT) initiative [3] 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)[4], 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 [6] and, therefore, would have needed to be provisioned through other out-of-band means, such as human-only-readable 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 and interact in the hypermedia environment.

2.1.2 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., [11–14]). For example, an approach that employs affordances as relationships in the con-

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

3 https://www.w3.org/WoT/

4 https://www.w3.org/TR/wot-thing-description/

text of affordance-based simulated agent environments comes from the work of Klügl and Timpf [13]. 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, and it reflects 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 perception could rely on the popularity of an affordance estimated by the designer of the environment. This would require that the designer adopts a user-centered perspective for designing perceptual information about affordances by considering specific types of agents that will inhabit the environment.

In [14], Papasimeon uses the notion of "annotation", which is a label put in the environment, and identifies an affordance as a special class of annotation. The author ascribes to annotations, i.e. to affordances, the following properties:

- 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 [7]. Therefore, perceptual information is not an affordance but a signifier - a cue for conveying an affordance[5]. 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 consequently, decouple the relational and objective aspects of affordance-driven interaction: Affordances remain relational, while signifiers can be designed on top carrying objective semantics.

In another approach, Joo et al. [11] utilize an affordance-based interaction formalisation [15] that considers affordances as properties of the environment. By bringing affordances to the computational space, the authors aim to more realistically simulate 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

[5] The distinction between affordances and perceptual information about affordances had also been established by William Gaver [6].

more like signifiers rather than affordances, since they are modelled and calculated based on affordance instances.

In his 2001 article [12], Raubal also studies the problem of wayfinding for agents, but 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 situated in the environment – an agent is able to reach its destination without constructing a mental model of a potentially dynamic or nondeterministic environment. Since these signs are perceivable and communicate appropriate behavior to an agent, they are signifiers according to Norman's definition. These signifiers can be added to the environment, and provided that they are designed in an agent-centered manner (i.e., they relate to agents' goals, per Papasimeon), they can be used to indicate personalized information to an agent in order to let that agent achieve its desired goal.

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 [2]. 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.

2.2.1 Exploiting affordances in human-computer interaction

Goal-driven behavior in affordance-based systems differ 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"[4]. 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, they are useful 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.

In a Web environment, hypermedia-driven interaction enables the provisioning of explicit information about an affordance — what it is for, and how it can be exploited. The way hypermedia is presented to humans (e.g., with a specific text color, label, entry field, 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 simply by interacting with the perceived affordances (i.e., the presented text or other media), while the browser is responsible for interpreting the provided information about *how to* use 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. Thus, interaction can be viewed as being affordance-driven, and, in this case, hypermedia-driven due to HATEOAS.

Exploiting affordances in a hypermedia-driven manner, relieves human agents from planning ahead every step towards achieving their goals, since they can incrementally determine their next action after each affordance exploitation. This is possible because *how relevant* the newly presented set of hypermedia is (so as to be conveyed) is (implicitly) taken into consideration by the environmental entity offering the affordances. This means that the origin server (and, optionally, any intermediaries) is responsible for computing which affordances should be conveyed based on the available run-time context; conventionally, based on the server's state, but also based on agent-related information (e.g., acquired in the context of Web sessions [16]) – and hence, that the *designer* of the hypermedia environment should consider the expected run-time state and goals of typical users at design time. Additionally, relevance can be estimated on the agent side, although initiated by the server and in a way hidden from the agent itself, e.g., through Web cookies [16] that hold information about the agent's past interactions. The more hypermedia-driven interaction is directed by the agent-environment situation, the closer it lies to affordance-driven interaction, thus, reducing the need for advanced reasoning and planning, and eventually, satisfying human agents' *opportunistic* goals and intentions effectively and efficiently: Don Norman considers as *opportunistic* those tasks, goals, and intentions that take advantage of the available run-time context, and, as a consequence, minimize the required mental effort and lay the foundations for creative solutions [4].

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* [17]. 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 [18] 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 [19]).

2.2.2 Exploiting affordances in multi-agent systems

Different methods for exploiting affordances in goal-driven interactions are also encountered in related work in MAS. In [11], 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 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 [20], 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 [21], and affordance theory is used to model the affordance-effectivity pairs [15] that enable agents to make decisions for taking 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 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. [20] mainly separate the concerns between the BDI-based and affordance-based system components, Papasimeon designs an affordance-based BDI reasoning processor. Since in [14] 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's reasoning process, the agent achieves its affordance-related intentions by executing the corresponding actions.

2.2.3 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 [22, 23]). A hypermedia environment is an environment that can be explored and exploited through hypermedia. A Hypermedia MAS is characterized by three principles [22]:

- Uniform resource space: All entities in a Hypermedia MAS (e.g., agents and non-autonomous environmental entities) 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 (e.g., to interact) 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 (e.g. an information resource about how to interact).

These principles enable and complement HATEOAS for artificial agents – a bridge that is established in [1, 5] and that corresponds with the methods for representing and reasoning about action as examined in MAS research by formalizing signifiers in the context of Hypermedia MAS in [5]. The approach relies on agents that are situated in a Hypermedia MAS environment, and are provided with the affordances to explore it, starting from a single entry point, and navigating towards discovering, observing, and interpreting information about affordances. Based on the exposed information, agents can deliberate about the usefulness and the relevance of affordances in the environment, and use the provided hypermedia controls to exploit affordances towards achieving their objectives. Moreover, agents do not need to be the only entities devoting computational resources towards reasoning about the relevance of affordances. Non-autonomous entities in the environment can be used as services that contribute to this process, thus potentially dynamically shifting the burden from the agent, especially when efficient interactions are of interest in large-scale and affordance-rich MAS environments. Since agents and environmental entities are modelled as resources, their run-time context could be observed by entities dedicated to handle such information towards reasoning on the agent-environment situation and managing the exposure of affordance-related information. To this end, the authors in [5] provide a formalization of a Signifier Exposure Mechanism that manages the contextual exposure of signifiers in hypermedia environments that is compatible with the present work as well as with [1].

In the following, on the basis of the broad foundation that we established across HCI and MAS research in this section, we present a conceptual model for representing interaction possibilities through signifiers in MAS, as well as a more abstract definition of the Signifier Exposure Mechanism. Our approach aims to highlight fundamental features of exploiting signifiers as a first-class abstraction, and to provide a foundation for the exploration and development of more specific models, such as the one presented in [5].

3 Affordance-driven interaction in multi-agent systems

Agents need to exploit affordances towards achieving their goals. In order to enable agents to perceive affordances, a model for signifiers capturing information about the affordances present in the environment is proposed in Section 3.1. A Signifier Exposure Mechanism is developed in Section 3.2 to permit more fine-grained control about what signifiers are perceivable by an agent (e.g., to let an agent only perceive those signifiers that are relevant to it in a specific situation). In Section 3.3, an explanation of how signifiers and affordances can be used by the agents is given.

3.1 Modelling affordances and signifiers

A signifier should correspond to a description of an affordance. In order to create a signifier, we first identify the components used to describe an affordance. Then, we show how the description of an affordance can be integrated within a signifier that also provides some meta-data that can be used to indicate whether it is relevant to an agent or not.

Affordance
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, exploitable by an agent under certain conditions of the agent-environment system, and the exploitation of an affordance updates the conditions in the system. We consider that an affordance is defined by *preconditions* and *postconditions*, themselves built on the concepts of *state*, *objective* and *action*. These concepts are defined in the next paragraphs.

State
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.

Preconditions and Postconditions The preconditions and postconditions of affordances carry the same semantics as these concepts do for actions (or operators) in classical planning [24]: 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). In our approach, the difference is that such conditions remain attached to affordances through signifiers, and are hence decoupled from specific and grounded actions which are only encountered and executed at run time.

Objective
An objective is a state that describes a goal that an agent can achieve by exploiting an affordance. In contrast to the postcondition, the objective is not necessarily reached as soon as the affordance is exploited, but exploiting the affordance will bring the agent closer to achieving this objective. If an agent has a given objective, the agent can compare this objective to the objectives associated to the affordance. If the agent's objective is equal to (or overlaps with) the objective of an affordance, the agent can reason that exploiting the affordance may work towards achieving its objective, and, thus, it may decide to directly proceed to exploitation. Most importantly, the objective can be reasoned upon in consideration with the preconditions and postconditions of an affordance. In this case, the agent can proceed to plan its actions using only the set of affordances that relate to its objective. An affordance can be related to many objectives and the description may not be exhaustive.

Action
An action indicates how an agent can exploit 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[6] or CoAP[7].

Signifier
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 paragraphs.

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

[6] https://datatracker.ietf.org/doc/html/rfc2616

[7] https://datatracker.ietf.org/doc/html/rfc7252

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 with respect to a workspace as defined in the Agents & Artifacts meta-model [25]) or physical (e.g., with respect to geographical coordinates).

Creator
The creator of the signifier may be the designer of the environment, or an agent. Knowledge about the creator can be used to 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 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.

3.2 Perceiving affordances through the signifier exposure mechanism

Agents need to perceive signifiers in order to gain information of how they might achieve their goals through the affordances that are described by these signifiers. We refer to signifiers that bring an agent closer to its goal as *relevant* signifiers. Depending on the run-time situation, all relevant signifiers may be necessary to let an agent achieve its goal (e.g., to perform complex behavior or planning) or only one signifier might be necessary. Hence, 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 (SEM) can be added to the environment. A SEM is a mechanism in the environment that determines the signifiers that an agent is able to perceive. A helpful SEM will filter all available signifiers and send only those it estimates to be relevant to the agent. The filtering can be done with the information that the SEM 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 SEM—indeed, the salience represents an a priori level of relevance that is independent of the agent's run-time situation. Since a salient signifier is more likely relevant than one that is not, salient signifiers should be given higher priority.

However, using information only from the content of the signifiers and the state of the environment may not be enough to permit the effective selection of agent-relevant signifiers. In addition, the agent's preferences and abilities should be taken into account. Here, with preferences we refer to the objectives that an agent aims to accomplish, meaning that signifiers may only be deemed relevant if they contribute to the achievement of these objectives. Abilities aim to capture the qualities of an agent within the agent-environment relationship, that determine whether an affordance is exploitable by the agent, and therefore, whether a signifier is relevant to the agent. For example, abilities may refer to an agent's knowledge of abstractions and processes within a specific domain, or the methods and mechanisms employed by the agent to reason about actions. Domain-specific knowledge may limit the set of relevant signifiers to those utilizing models specific to a domain such as models of industrial or healthcare processes. Methods and mechanisms for reasoning about action can also guide the filtering process, for example, enabling an agent to perceive an enlarged set of signifiers if it is capable of performing automated planning. As a result, information about an agent's preferences and abilities may be greatly useful to determine whether the agent can exploit an affordance or interpret information about it, and whether the affordance supports the agent in achieving its goal.

It is therefore efficient to take into account the agent's preferences and abilities when evaluating the relevance of signifiers to an agent, and performing, accordingly, the filtering of signifiers. However, the SEM does not have direct access to them because they are only internal to the agent. In order to allow the agent to share information about its preferences and abilities with the SEM, we propose that this information should be made accessible inside the environment. To do so, an agent may create an *agent profile*, located inside the environment, to indicate its preferences and abilities (including the agent's knowledge, such as knowledge of a password to access a given resource), and any information that the agent wants to make available about itself to other agents and to the environment's SEM. If the SEM is given access to this profile, it can use its content to improve the filtering of signifiers. For example, if the agent's abilities prevent 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. Nevertheless, the decision regarding the inclusion of information in an agent profile and the degree of flexibility in the filtering applied by the SEM ultimately rests with the agent. If an agent chooses to configure the mechanism to disable filtering, it can independently evaluate signifiers based on its internal state and the fully exposed set of signifiers, including the revealed information about the supported objectives or abilities. This approach preserves the agent's autonomy from the environment as it avoids missing any signifiers that may get filtered by another component. However, it comes at the expense of increased processing time for the agent, particularly in large-scale environments. On the other hand, in case the filtering is delegated to an SEM in the environment, autonomy of the agent will be constrained for receiving support in the filtering task.

In order to determine which signifiers would be relevant to the agent, the SEM compares the content of the Agent Profile with the signifiers that relate to entities offering affordances in the environment. We consider that each entity offering affordances is related to an *Entity Profile*, i.e. a document, located within the environment, that indicates information that the entity makes available about itself as well as the signifiers it can expose. Therefore, the SEM can determine which signifiers should be exposed to the agent by comparing the Agent Profile with the Entity Profile (see Fig. 1). As a result, the SEM provides environment-driven support for interaction, serving as a complement to agent-driven approaches. We argue that this combination unlocks the potential for agents to dynamically distribute the workload of driving interaction between themselves and their environment. Furthermore, independently of the filtering level that is applied, 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.

Finally, signifiers can be created and published not only by environment designers but also by agent designers and potentially the agents themselves. In the latter case, agents can take advantage of signifiers as a first-class contruct in the MAS to enrich their environment, and share information about interaction possibilities with other agents. For example, in Fig. 1, an agent updates the profile of the entity offering affordances, thereby introducing a new signifier into the hypermedia environment. This signifier can then be utilized by the same agent for future interactions or discovered by other agents within the system.

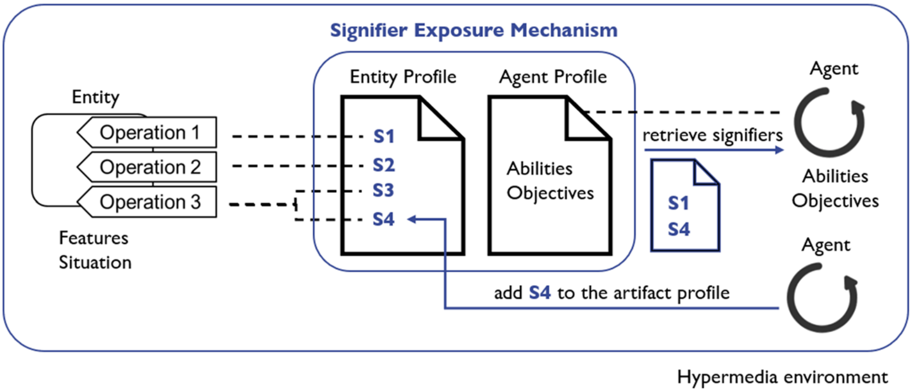

Fig. 1 The Signifier Exposure Mechanism exposes signifiers that reveal information about interaction possibilities offered to autonomous agents. Exposure is determined based on the relevance of signifiers to agents' abilities and objectives. Any information that is useful for relevance evaluation, including signifiers, objectives, abilities, and preferred relevance thresholds, can be dynamically published in the environment by agents to assist the dissemination and management of signifiers

3.3 Exploiting affordances in goal-driven behavior through signifiers

In classical planning, a plan is constructed as a sequence of actions, characterized by their preconditions and postconditions, so that the first action's precondition is satisfied when the plan is used, each following action's precondition is satisfied after the execution of the previous action, and the final action leads to a goal state. The construction of such a plan requires knowledge about the actions that are available in the environment. However, this knowledge may not be a priori available to the agent. Besides, if the environment is dynamic and can potentially be changed by other agents, the state of the environment may change independently of the agent's actions thus making the plan unusable.

However, it is still possible for an agent to act without a plan if it can find out signifiers in the environment. Signifiers can indicate which affordances are available for the agent, whether they can help the agent achieve its goal, and how to exploit them. Therefore, signifiers can provide some information that can be used by the agent to decide which action to perform and how to perform it. For example, the agent may decide to perceive signifiers, then choose the proper action to perform based on the signifiers and the goal, then check whether the goal is reached or not. If not, the agent continues pursuing the goal. The agent should therefore be able to execute different actions in order to: check whether its goal is achieved or not, perceive signifiers, select an affordance to exploit based on these signifiers in order to bring the agent closer to its goal. The algorithm that the agent follows to achieve its goal is therefore the following one:

Listing 1 Algorithm indicating how the agent can select the affordances to use

```
updateProfile(goal) //The agent updates its profile
    with its goal.
while (not achieved(goal)):
    signifiers = perceive()
    chosenAffordance = chooseBest(signifiers, goal)
    act(chosenAffordances)
```

Signifiers can therefore be used by agents in complex and dynamic environments where planning would be impossible. As a result, signifiers increase the autonomy of the agent by enabling the agent to act even when a plan is not available.

However, apart from enabling agents to act opportunistically based on the agent-environment context, signifiers can additionally be used to perform classical planning based on the preconditions and postconditions that they capture. After synthesizing an initial plan, signifiers can facilitate agents to act based on the synthesized plan or replan their actions. Acting is enabled since signifiers reveal information about how the actions of the plan can be performed, i.e. they reveal the commands to be executed for performing the action (e.g., in the form of requests in a specified protocol like HTTP). For example, even if a signifier that was used during planning is no longer available, commands may still be found by perceiving another signifier for an action with the same preconditions and postconditions. At the same time, if a plan is partial or fails during execution, signifiers can again be perceived and indicate the new actions available at run time, that can be used for replanning. Such actions may be different from the ones that were available at planning time and this is why the new plan may succeed where the initial plan failed. An example of how a signifier can be designed for a STRIPS planner is presented in [5], where the authors use a PDDL ontology to define a signifier [8] exposed by a robotic arm entity which offers the affordance of closing the arm's gripper. This signifier reveals information about how to close the gripper, as well as the preconditions and postconditions of the related action.

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[9] [22], and the creation of a JaCaMo application [26] containing agents able to use hypermedia artifacts with signifiers on Yggdrasil in Section 4.1. We tested our implementation with a maze scenario developed on Yggdrasil and the JaCaMo project in Section 4.2.

4.1 Implementation

Three ontologies are developed: an ontology to model signifiers[10], an ontology to model agent profiles[11], and an ontology to model hypermedia actions[12]. A Java library is created, based on these ontologies, in order to create and process signifiers based on these ontologies. 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.

Yggdrasil is a framework coded in Java, based on the framework `Vert.x`[13], 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 available

[8] In [5], the example signifier is defined based on a model that slightly differs from the model presented in this paper. For example, the authors do not explicitly incorporate preconditions and postconditions into their model, though they do acknowledge the potential for these elements as an extension.

[9] https://github.com/Interactions-HSG/yggdrasil

[10] https://w3id.org/interactions/ontologies/signifiers/v1

[11] https://w3id.org/interactions/ontologies/profile/v1

[12] https://w3id.org/interactions/ontologies/hypermedia/v1

[13] https://vertx.io/

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[14], 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.

In this scenario, the agent is in a maze (Fig. 2) 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 the agent to cross the maze efficiently.

Listing 2 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>.

_: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]".
```

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

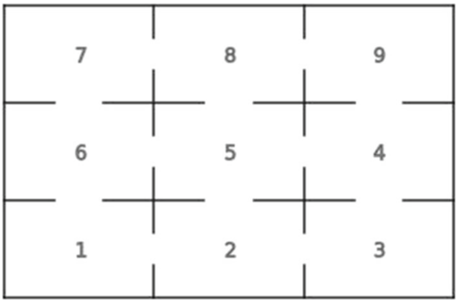

Fig. 2 Representation of the maze used in the scenario

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 has the objective to go to room 9. The agent may also add an upper limit for the number of signifiers 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 2) indicates an affordance that brings the agent closer to its destination. For example, in room 2, the signifiers shown are for moving from room 2 to room 5 or room 3, but not to room 1 because this would not be complementary to the agent's objective of reaching room 9.

A second scenario is developed to illustrate how signifiers can enable a stigmergic interaction between agents. This scenario features two agents, each one remains unaware of the existence of the other agent. The first agent has an algorithm to find an optimal path in the maze from the entrance (room 1) to the exit (room 9) and is able to add signifiers to the maze in order to indicate this optimal path. After the first agent has found out an optimal path in the maze and added the proper signifiers, the second agent enters the maze at room 1 and follows the signifiers in order to reach room 9.

4.3 Evaluation

The scenario illustrates different benefits 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 actually 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, or the path could even be longer if an agent can visit one room multiple 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 act 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 fewer signifiers, which therefore which therefore decreases the consumption of the agent's available memory. Consuming less of the available 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.
- The scenario involving two agents show that the introduction of signifiers allows for stigmergic interactions between agents.

5 Discussion

Human agents 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 to reach the exit if exit signs are present in the building—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 use any signifier provided they also know the ontologies used to describe the actions in the environment. In our implementation, we define hypermedia actions that represent the HTTP requests that an agent needs to perform in order to exploit 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 expose to the agent only the signifiers it needs to use. Therefore, the Signifier Exposure Mechanism can reduce the processing time of signifiers by the agents.

In hypermedia systems, the HATEOAS principle enables guidance through following hyperlinks. However, this basic guidance does not take into account the agent's goals and abilities. The SEM can be used to provide a more informed guidance based on these goals and abilities, indicated in an agent profile. Additionally, the SEM could enable agents to add signifiers in the set of signifiers that the SEM can display, therefore enabling potentially better guidance for other agents, and decoupling the designer of signifiers from the designer of artifacts. This feature is necessary to make the SEM useful in dynamic and scalable environments where a single designer is not able to create signifiers for the full system, which can be extended at run time. However, such additional signifiers, created by agents, might be untrustworthy or incorrect. We consider that the SEM could be integrated with some recommender system, implemented with machine learning techniques [27], to learn the relevance of added signifiers and adapt signifier exposure accordingly. Agents may also need to provide feedback on the signifiers they are shown to enable the SEM to evaluate the better the relevance of the signifiers. Enabling agents to add signifiers to the SEM and provide feedback on signifiers may lead to trust and privacy issues. However,agents remain free to decide whether to trust or not any single signifier that they perceive and remain in control of the feedback they provide to the SEM, thus mitigating the envisioned issues.

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.

In the first scenario, we show that an agent does not need to possess a full plan to escape a maze but can rely on information provided by signifiers at run time. In the second scenario, we show how signifiers and the Signifier Exposure Mechanism can be employed as a means for featuring cognitive stigmergy [28] in a Multi-Agent System. These scenarios offer a concise illustration of signifiers as a shared abstraction for both environment designers and artificial agents: The concept of signifiers inherently supports the dynamic publication and context-aware exposure of information about interaction possibilities. These features can lend themselves to more complex use cases, including cases where agents wish to dynamically delegate or share reasoning workload, lack planning capabilities, or aim to enrich their environment for assisting themselves in the future or for coordinating with other agents.

6 Conclusions

In this paper, we reviewed approaches and methods aimed at enabling the conveyance of affordances from the environment, and the exploitation of these affordances by agents in socio-technical systems. Building upon this conceptual foundation, we proposed the enrichment of the environment in Multi-Agent Systems with signifier, that provide artificial agents with information about the affordances they can exploit to achieve their goals. 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 enable agents to perceive only the signifiers that are relevant to them, the environment features 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 with 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 serves as a proof-of-concept demonstrating how signifiers can enrich the environment in a Multi-Agent System, and how their context-aware and environment-driven exposure can support agents' interactions.

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.

Appendix A Extension notes for "signifiers for affordance-driven multi-agent systems"

A.1 Revision and extension of Section 2 "agent-environment interaction in socio-technical systems"

Section 2 was revised to provide a more detailed description of the "Hypermedia as the Engine of Application State" principle as part of the Representational State Transfer architectural style (see Section 2.1.1). The revision aims at establishing a clearer view of how signifiers are used in hypermedia-driven interaction based on the Web architecture.

Additionally, the description of the means through which human agents exploit affordances on the Web in a goal-driven manner was extended (see Section 2.2.1). Through this section, we now elaborate on how affordances are strategically presented to human agents based on the run-time agent-environment context towards enabling agents to more efficiently achieve their goals on the Web environment.

Finally, the background knowledge on Hypermedia Multi-Agent Systems was revised and extended to better articulate why this class of Multi-Agent Systems offers an appropriate infrastructure a) for exploiting signifiers in agent-to-environment interactions, and, b) for pointing out the advantages of adjusting the exposure of signifiers based on the agent-environment context in dynamic and affordance-rich Multi-Agent Systems (see Section 2.2.3).

A.2 Revision and extension of Section 3 "affordance-driven interaction in multi-agent systems"

Section 3.2 was renamed to "Perceiving Affordances through the Signifier Exposure Mechanism".

Section 3 was revised to improve readability and increase the clarity of the presented definitions. Specifically, references to state implications in the definition of a state were removed because we considered them unnecessary (see Section 3.1). The definition of the Signifier Exposure Mechanism, was updated to present the notion in a clearer manner, and a new definition, the notion of Agent Profile which is used throughout our work (see Section 3.2) was defined. Figure 1 was added to represent the Signifier Exposure Mechanism. The definitions have been integrated within the paragraphs they refer to instead of being separated from them.

Section 3.3 was refactored and extended. The title of the subsection was changed from "Usage of Affordances in Agent Plans" to "Exploiting Affordances in Goal-driven Behavior through Signifiers" to broaden the scope of the subsection, describe plans as representing one type of goal-directed behavior, and to indicate that the affordances can be exploited by the information provided by signifiers. Finally, the discussion concerning classical planning is extended. The process by which an agent can use signifiers as an alternative to classical planning is explained in greater detail, and an algorithm in pseudo-code is provided to illustrate the process. There is also a discussion on how an agent using a classical planner could benefit from perceiving signifiers.

A.3 Revision and extension of Section 4 "implementation and experience"

A second scenario was added to demonstrate the use of signifiers in a multi-agent context (see Section 4.2). Also, the evaluation of the implementation was updated: the bullet point concerning the computational cost of using signifiers was reformulated, and the bullet point 4 was added concerning the benefits of signifiers in enabling stigmergic interaction as illustrated by the second scenario.

A.4 Revision and extension of Section 5 "discussion"

The discussion was augmented with a discussion on signifiers and HATEOAS, as well as a discussion of the benefits of signifiers and the Signifier Exposure Mechanism as illustrated by the two scenarios. The benefits and challenges associated with enabling agents to add signifiers to the Signifier Exposure Mechanism are better explained.

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*).

Funding Open access funding provided by University of St.Gallen.

Availability of Data and Materials Data sharing not applicable to this article as no datasets were generated or analysed for this published work.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Lemée, J., Vachtsevanou, D., Mayer, S., Ciortea, A.: Signifiers for Affordance-driven Multi-Agent Systems. Paper presented at the 10th International Workshop on Engineering Multi-Agent Systems 9–10, Auckland, New Zealand (2022)
2. Wooldridge, M.: Intelligent Agents. MIT Press (2013)
3. Gibson, J.J.: The Ecological Approach to Visual Perception, Classic Press, Psychol (2014)
4. Norman, D.: The Design of Everyday Things: Revised and Expanded Edition. Basic Books (2013)
5. Vachtsevanou, D., Ciortea, A., Mayer, S., Lemée, J.: Signifiers as a First-class Abstraction in Hypermedia Multi-Agent Systems. In: Proceedings of the 22nd international conference on Autonomous Agents and Muliagent Systems, pp. 1200–1208 (2023)

6. Gaver, W.W.: Technology Affordances. In: Proceedings of the SIGCHI Conference on Human factors in Computing Systems, pp. 79–84 (1991)
7. Norman, D.A.: THE WAY I SEE IT Signifiers. Not Affordances. Interactions **15**(6), 18–19 (2008)
8. Şahin, E., Cakmak, M., Doğar, M.R., Uğur, E., Üçoluk, G.: To Afford or Not to Afford: A New Formalization of Affordances Toward Affordance-based Robot Control. Adapt. Behav. **15**(4), 447–472 (2007)
9. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine (2000)
10. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Sci. Am. **284**(5), 34–43 (2001)
11. Joo, J., Kim, N., Wysk, R.A., Rothrock, L., Son, Y.J., Oh, Yg. et al.: Agent-based Simulation of Affordance-based Human Behaviors in Emergency Evacuation. Simul. Model. Pract. Theory. **32**, 99–115 (2013)
12. Raubal, M.: Ontology and Epistemology for Agent-based Wayfinding Simulation. Int. J. Geogr. Inf. Syst. **15**(7), 653–665 (2001)
13. Klügl, F., Timpf, S.: Towards More Explicit Interaction Modelling in Agent-Based Simulation Using Affordance Schemata. In: German Conference on AI (Künstliche Intelligenz). Springer, pp. 324–337 (2021)
14. Papasimeon, M.: Modelling Agent-Environment Interaction in Multi-Agent Simulations with Affordances. Defence Science and Technology, Edinburgh, Australia (2010)
15. Kim, N., Joo, J., Rothrock, L., Wysk, R., Son, Y.J.: Human Behavioral Simulation Using Affordance-based Agent Model. In: International conference on human-computer interaction. Springer, pp. 368–377 (2011)
16. Barth, A.: HTTP State Management Mechanism. Internet Engineer, Task Force (IETF) (2011)
17. 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, pp. 47–69 (2020)
18. Verborgh, R., Haerinck, V., Steiner, T., Van Deursen, D., Van Hoecke, S., De Roo, J. et al.: Functional Composition of Sensor Web APIs. In: SSN, pp. 65–80 (2012)
19. Kovatsch, M., Hassan, Y.N., Mayer, S.: Practical Semantics for the Internet of Things: Physical States, Device Mashups, and Open Questions. In: IOT 2015. IEEE, pp. 54–61 (2015)
20. Joo, J.: Perception and BDI Reasoning Based Agent Model for Human Behavior Simulation in Complex System. In: International Conference on Human-Computer Interaction. Springer, pp. 62–71 (2013)
21. BDI Agents:Rao, A.S., Georgeff, M.P., et al.: From Theory to Practice. In: Icmas. **95**, 312–319 (1995)
22. Ciortea, A., Boissier, O., Ricci, A.: Engineering World-Wide Multi-Agent Systems with Hypermedia. In: International Workshop on Engineering Multi-Agent Systems. Springer, pp. 285–301 (2018)
23. 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)
24. Ghallab, M., Nau, D., Traverso, P.: Automated Planning and Acting. Cambridge University Press (2016)
25. Ricci, A., Piunti, M., Viroli, M.: Environment Programming in Multi-Agent Systems: An Artifact-based Perspective. Auton. Agent. Multi-Agent Syst. **23**(2), 158–192 (2011)
26. Boissier, O., Bordini, R., Hübner, J., Ricci, A., Santi, A.: Multi-Agent Oriented Programming with JaCaMo. In: Science of Computer Programming, pp. 747–761 (2013)
27. Portugal, I., Alencar, P., Cowan, D.: The use of machine learning algorithms in recommender systems: A systematic review. Expert Syst. Appl. **97**, 205–227 (2018)
28. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive stigmergy: Towards a framework based on agents and artifacts. In: Environments for Multi-Agent Systems III: Third International Workshop, E4MAS 2006, Hakodate, Japan, May 8, 2006, Selected Revised and Invited Papers 3. Springer, pp. 124–140 (2007)