Pody: a Solid-based approach to embody agents in Web-based Multi-Agent-Systems

Antoine Zimmermann, Andrei Ciortea, Catherine Faron, Eoin O'Neill, María Poveda-Villalón

Pody: a Solid-based Approach to Embody Agents in Web-based Multi-Agent-Systems

Antoine Zimmermann[1], Andrei Ciortea[2,3], Catherine Faron[3], Eoin O'Neill[4], and María Poveda-Villalón[5]

[1] Mines Saint-Étienne, Univ. Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, Saint-Étienne, France
antoine.zimmermann@emse.fr
[2] University of St. Gallen, St. Gallen, Switzerland
andrei.ciortea@unisg.ch
[3] Université Côte d'Azur, CNRS, Inria, I3S, Sophia Antipolis, France
faron@i3s.unice.fr
[4] University College Dublin, Dublin, Ireland
eoin.o-neill.3@ucdconnect.ie
[5] Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
mpoveda@fi.upm.es

Abstract. In this paper we discuss the problem of situatedness for agents perceiving and acting on the Web (namely, "Web agents"). Assuming Web agents are *embodied* on the World Wide Web, then we must define what is a Web agent's *body*. We first provide an abstract definition of a Web agent's body in terms of what it should comprise. Then we propose a concrete definition of it relying on Solid, a recent Web technology for Social Linked Data: we implement a Web agent's body as a data pod. Consequently, we coin the term *pody* to refer to the Web entity that embodies an agent on the Web with Solid. This paper summarises the findings of a working group from the *Dagstuhl Seminar 23081: Agents on the Web* (February 19-24, 2023).

Keywords: MAS · Semantic Web · Solid· Embodiment · Situatedness

1 Introduction

Situatedness and *embodiment* are key notions in research on intelligent agents. The dominant view is that intelligent, rational behaviour is closely related to the environment an agent occupies and is not disembodied [17]. This view emerged in the late '80s in close relationship with research on intelligent robots [11], which are naturally situated and embodied in a physical environment. The complexity of virtual environments, such as the Web, now rivals that of physical environments. Furthermore, with the recent standardisation of the Web of Things at the W3C and the IETF, the Web now extends to the physical world – and thus becomes a uniform hypermedia fabric that interconnects virtual and physical environments. This evolution unlocks new practical use cases for intelligent agents on the Web, that need to be situated and embodied in their environment. This vision that can be traced back to the early days of the Web [6].

In this paper, we discuss how Web agents can be embodied into the Web, both at an abstract level and concretely using Web standards and technologies. In a nutshell, we envision a Web agent's body as a collection of Web resources and Web interfaces that are attached to the identity of the agent. The Web agent's body allows the agent to participate in collective work as part of a multi-agent system (MAS) on the Web: to perceive and actuate Web resources (including Web-enabled devices), to be discovered and perceived by other agents, to participate in organisations, to communicate with other agents, etc. We illustrate this vision through a concrete example of Web agents embodiement using Solid pods, the core concept and technology from Sir Tim Berners-Lee's project for **So**cial **Li**nked **D**ata – an initiative to preserve the decentralised nature of the Web and to radically decentralise personal data. In particular, this enables to seamlessly address MAS use cases where a strong emphasis on ownership of the agents' personal data and resources is needed.

The paper is organized as follows: We first present in Section 2 the context in which our proposal arose. Then we present in Section 3 our vision of how agents should be situated and embodied on the Web, independently of the technologies used. Finally we show in Section 4 how this can be implemented using Solid. In the end, we discuss in Section 5 what other abstractions would be needed to articulate *podies* with other essential dimensions of Web-based MAS and we conclude in Section 6.

2 Background

In this section, we first discuss the notions of situatedness and embodiment in Artificial Intelligence – and, in particular, in MAS engineering (Section 2.1). Then we provide an overview of the main Semantic Web concepts, principles, and technologies on which the implementation of our proposal is relying (Section 2.2). Finally, we present Solid, the key technology at the center of our proposal (Section 2.3).

2.1 Situatedness and Embodiment in Multi-Agent Systems

In the mid-80s, a new view emerged in the research field of intelligent agents: an agent is considered situated in its environment, in the sense that it is directly connected to its problem domain through sensors and actuators, and it can effect changes in this domain through actuators [11]. This view contrasted prior views in AI research, in which an agent would typically amount to a program to which a formal specification of a problem is provided as input – and then the program returns a result.

[6] See the keynote of Sir Tim Berners-Lee at the First International Conference on the World Wide Web (WWW'94): `https://videos.cern.ch/record/2671957`

The notion of *situatedness* originated from research on mobile robots, with Brooks being one of its main originators [4]. It is now generally accepted for any system that needs to autonomously fulfill its design objectives in a dynamic, unpredictable environment – be it physical or virtual [11]. Most definitions of what is an *intelligent* or *autonomous agent* are centered around this notion of *situatedness* (e.g., see [6] for a detailed discussion of various definitions).

Another notion closely related to *situatedness* is the *embodiment* of an agent. In [3], Brooks defined this notion to articulate that robots have bodies and "their actions are part of a dynamic with the world" (e.g., their actions have immediate feedback on their own perception). Close to situatedness, this notion of embodiment originally applied to mobile robots can be extended to agents in virtual environments.

Most notably, in the Agents&Artifacts (A&A) metamodel [13] for MAS engineering, agents are situated in *workspaces* where they are embodied through *body artifacts*. A body artifact holds an agent's context within a workspace: it allows the agent to perceive and act within the workspace, and it allows other agents situated in the same workspace to perceive and interact with the agent. An agent holds a body artifact in each workspace it is a part of. From an engineering viewpoint, this separation of concerns between an agent's mind and its body artifact allows heterogeneous agents (e.g., using different architectures or frameworks) to be reified within the same workspace in a uniform way.

2.2 A Web for Machines

In 2001, the Semantic Web was defined as an extension of the Web relying on new models and technologies to provide structure and meaning to the content available on the Web [2]. The Semantic Web relies on the Resource Description Framework (RDF), a graph model to structure data by expressing relations between entities, and on RDF Schema and the Ontology Web Language to represent the ontologies used in RDF graphs, thus providing semantics to them.

Early research on the Semantic Web was mostly focused on ontology engineering and knowledge representation, but in 2006 Tim Berners-Lee introduced the Linked Data principles [1], that are summarised as follows: 1) use URIs to name things; 2) use HTTP URIs so that things' names can be looked up; 3) describe things using standards (RDF) so useful information is provided for URIs; and 4) include links to other URIs in things descriptions.

Ontologies and linked data together provide the means by which an agent can reliably interpret resources described on the Web, whether they are digital resources or real-world resources. Additionally, with links, a Web resource leads to other resources, and so forth, so as to make agents aware of the environment that the Web constitutes. Some standardised ontologies also define, in their specification, conformance obligations that say how to operate with resources described using them. For instance, the W3C Thing Description standard [9] provides both an ontology to describe possible interactions with *things* on the Web, and the way those descriptions can be leveraged to operate these things.

2.3 Solid: Social Linked Data

Solid is a project launched by Tim Berners-Lee in reaction to the growing centralisation of Web platforms that collect more and more personal data. Instead, Solid aims at decentralising personal data management in such a way that Web users regain ownership and control over their data. At the core of Solid technologies, there is the Solid *pod* (**p**ersonal **o**nline **d**ata store) that hosts the user's data and is implemented as a Linked Data Platform [16] with access control on top of it. Pods are mostly used to provide data to online applications, such as social platforms, that are granted access by the pod's owner. This way, not only the data are externalised from Web platforms, but also the same identity, described inside the pod, can be reused across multiple applications.

Identity is managed using a customised protocol based on WebID [14] that allows one to retrieve credentials from a URI that not only identifies the user (as an account login) but also dereferences to the owner's data pod, thus enabling applications to get appropriate data from the user.

Solid pods can host any kind of data but are designed in particular to easily manage RDF datasets with fine-grained read/write operations. Overall, the Solid Protocol [5] specifies authentication, storage, access control, and interactions that must be implemented by Solid pods and Solid platforms in order to interoperate with each others and with applications that builds on them.

3 Embodiment and Situatedness of Agents on the Web

The **situatedness** of an agent, as introduced in Section 2.1, refers to the relationship that exists between the agent and its environment. In order for an agent to be situated in its environment, it must have the ability to perceive and act on it. In the case of a *Web agent* addressed in this paper, the environment comprises the Web, and the interactions are the basic interaction protocols defined for the Web. The minimum requirement for a *Web agent* is the ability to interact with hypermedia resources on the Web.

The **embodiment** of an agent on the Web requires a representation of the agent to exist within the Web. We define the *embodiment* of a *Web agent* as the composite set of resources it exposes within a Web-based hypermedia environment, including any (semantic) descriptions of such resources. A defining characteristic of an agent's embodiment in the Web is that the set of resources constituting an agent's **Web body** is innately tied to the agent's identity: the agent may be acting through its Web body, and other agents observing the body would assume that the entity controlling and acting through the body is indeed the reified agent. An agent could have multiple Web bodies, each representing the agent's context in a specific hypermedia environment. This paper posits that the minimum requirement for an "embodied Web agent" is a hypermedia resource that provides the semantically defined abstraction of an **Agent Description**, which may link to any Web bodies the agent might have. This is the top-level abstraction that describes the agent's resources on the Web—and the entry point into what we are considering to be the *embodiment* of the agent.

In order to facilitate interactions within a Web-based MAS, additional abstractions may be defined to provide the necessary contextual information, such as: **Communication Interfaces**, **Preferences**, **Goals**, or **Beliefs** — which are important abstractions for supporting collaboration and coordination in MAS (e.g., see [15,12]. Such abstractions may be shared based on conditional access. The **Communication Interface** abstraction is the element of the agent's embodiment that facilitates interaction between agents and allows for an agent to become an entity directly accessible within the Web. The **Preferences** abstraction provides information such as an agent's preferred methods of interaction, but it is not limited to that. It can also be a domain-specific abstraction that defines the agent's preferred environmental state or any other preference with regard to the agent's embodiment in a particular environment.

If an agent has an explicit internal representation of its goals, the **Goals** abstraction would allow the agent to expose a set of goals. The agent may not necessarily be actively pursuing these goals but by merely exposing a set of goals publicly as a resource, the agent can have an effect on other agents within the system. This can result in benevolence between agents or agents acting in the disinterest of other agents within the system, depending on the context and implementation. Similar to the **Goals** abstraction, if an agent represents its knowledge of the world in terms of beliefs (e.g., as it is the case for BDI agents [7]), the **Beliefs** abstraction exposes a set of beliefs as Web resources, so that other agents can query the supposed beliefs of the agent. Additionally, the publicly available beliefs of the agent may or may not be beliefs that the agent maintains, but can be an attempt to influence the environment state through the actions of other agents that inhabit it.

4 Podies: Solid Pods Implementing Web Agents' Bodies

In this section, we show how Solid pods can be used to implement the abstractions introduced in Section 3. The Solid protocol states that "an agent is a person, social entity, or software identified by a URI; e.g., a WebID denotes an agent". We then assume that such a URI would dereference to an entry point for the data pod of the agent, where an **Agent Description** would be provided as an RDF graph, in addition to the mandatory credentials for authenticating the agent. We call the Solid pod implementing an agent's Web body a *pody*. Listing 1.1 shows an example **Agent Description** for a self-driving bus agent's pody. It identifies the self-driving bus as an instance of the `foaf:Agent`[7], class from the Friend-of-a-Friend (FOAF) vocabulary (part of the Solid protocol) and it provides basic information about the agent (e.g., a name, a relevant image) and links to other resources that are part of the agent's Web body, namely: a **Communication Interface** in the form of a mailbox that can be used to contact the agent, and the agent's **Preferences**.

[7] See term definition: `http://xmlns.com/foaf/0.1/#term_Agent`

Listing 1.1. Example self-driving bus agent's pody: RDF representation of its **Agent Description** (in Turtle).

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix pody: <http://someuri.ext/pody/> .
@prefix solid: <http://www.w3.org/ns/solid/terms#> .

<#agent-desc> a foaf:PersonalProfileDocument ;
    foaf:primaryTopic <#webagent> .

<#webagent> a foaf:Agent ;
    foaf:name "Self-driving Bus 101" ;
    # Link to a communication interface (e.g., mailbox, news feed, etc.)
    pody:contact <mbox> ;
    # Link to preferences (entry point to different kinds of preferences)
    pody:preferences <pref> ;
    # Links to the OpenID Provider that will validate the authentication
    # (part of the Solid protocol)
    solid:oidcIssuer <https://oidc.example> ;
    # Links to a relevant image of the bus
    foaf:img <images/picture.jpg> .
```

Listing 1.2 shows a sample description of the bus agent's mailbox. In this example, the mailbox is, in fact, a Web service that can be used to contact the bus agent – and the service is described by a W3C WoT Thing Description. The mailbox's Thing Description allows other agents to use the service based on an abstract semantic model of the mailbox (rather than having to hardcode the specific interface of the mailbox). Other similar approaches, such as Hydra [10], could be used to describe the interface of the mailbox.

Listing 1.2. WoT description of the self-driving bus agent's mailbox (in Turtle).

```
@prefix td: <https://www.w3.org/2019/wot/td#> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
@prefix pody: <http://someuri.ext/pody/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<mbox> a td:Thing;
  td:hasActionAffordance [
    a pody:SendDirectMessage ;
    td:name "send-mail";
    td:hasForm [
      hctl:hasTarget "https://domain.ext/mbox/inbox"^^xsd:anyURI
    ]
  ] .
```

Listing 1.3 shows a sample representation of the bus agent's **Preferences**. In this example, the preferences expose a basic access control policy using the Web Access Control[8] vocabulary (part of the Solid protocol). Other preferences could express, for instance, a prioritization of the **Communication Interfaces** exposed by the agent – similar to the preferred ordering of contact addresses in a FIPA Agent Identifier as defined by the FIPA Agent Management Ontology[9].

Listing 1.3. RDF description of the self-driving bus agent's preferences (in Turtle).

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .

<pref> acl:accessControl [
```

[8] https://solidproject.org/TR/wac

[9] See the FIPA Agent Management Specification for details: http://fipa.org/specs/fipa00023/SC00023K.html

```
    acl:accessTo <mbox> ;
    acl:agent <http://example.edu/p/Alice#Msc>,
     . <http://example.com/people/Mary/card#me> ;
    acl:mode acl:Read
] .
```

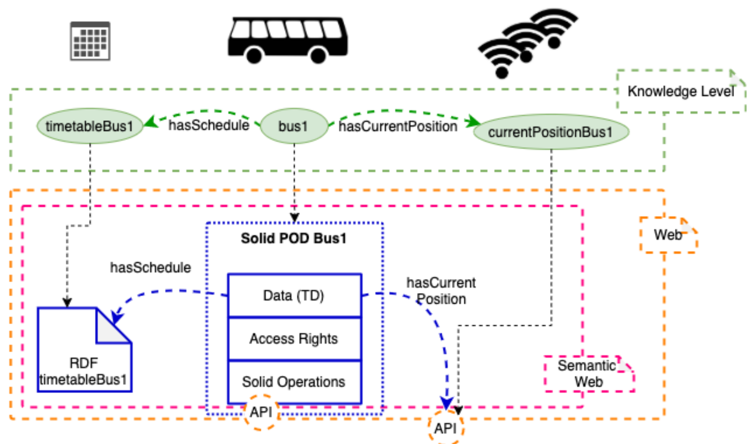

Fig. 1. Example bus agent embodied on the Web, its Solid pod implemented using Semantic Web models. The bus agent publishes on the Web its up-to-date position stored in its pody.

In addition to the resources described so far, the embodiment of the bus agent could include additional resources. A more elaborate illustration of this use case is shown in Figure 1. The bus agent could use its Solid pod, for instance, to publish an up-to-date schedule or its current position. Because such information is published under the bus agent's pody, other agents would assume that it is indeed the bus agent communication through its pody — similar to how a Twitter user would communicate updates via their Twitter account.

5 Discussion

Our proposal gives uniformity to how agents are embodied on the Web. The notion of *pody* makes use of technologies that are mostly based on standards, as well as work that is under active development by public organisations and companies. The use of Linked Data enforces uniform identification (with URIs), a common data model (with RDF), and a way of serendipitously exploring data, especially for what concerns agents, their means of communication, and their specificities.

With agents embodied in the Web through *podies*, we can envision how they can be situated and related to other dimensions of a multi-agent system. Other abstractions would have to be introduced to describe the Web counterpart of a physical location. We can assume that agents will cooperate on the Web within abstract areas that delimit the scope of their interaction and offer the required resources to address specific missions, goals, and endeavours. For instance, agents may collaborate in Github projects, with a repository acting as a workspace where they are situated. These "abstract areas" or workspaces can themselves be described in podies that would also offer interaction facilities, links to the agents situated in them, ways of taking roles, etc.

Additionally, agents cooperating in complex organisations, possibly with a mix of human beings, software agents, or robots, should be able to obtain organisational information, such as norms, regulation, and so forth, in a form that is easily machine-processable. Interestingly, existing Web ontologies already cover parts of these abstractions, and research communities are actively working on providing shared vocabularies that enable to precisely to describe these things.

6 Conclusion

Web agents can be embodied via a Solid pod that: 1) provides a recognisable identity to the agents acting/interacting on and via the Web; 2) provides a "shape" to the agent in the form of an agent description, materialised as an RDF graph; 3) provides an interface through which other agents can communicate with the embodied agent; 4) may optionally provide supporting features such as preferences, claimed goals and beliefs, all possibly represented using standards.

This paper posits a shape, and an implementation method, that may be used to represent the intelligent agents that inhabit the Web, the same intelligent agents that Hendler was querying the existence of in [8]. We see this as a step in the direction of allowing agent technologies to be utilized in a Web context consuming semantically enriched data and interacting in an ad-hoc fashion with heterogeneous, semantically described Web services in order to provide services and pursue and achieve goals of their own. By defining a standard abstract "shape" for a *Web agent*, using Web standard technologies, we introduce the possibility of cross-organisational interaction and collaboration.

The contribution of this paper is a vision that still requires a realisation in an actual MAS. We argue that this vision already shows the benefits for engineering Web-based MAS. Future work will determine, by experimentation, the feasability, usability, ease of development, scalability, and perhaps limitations for Web-based Multi-Agent Systems engineering.

Acknowledgement. We thank Alessandro Ricci and Jomi Hübner for our fruitful discussions during the Dagstuhl Seminar 23081 that led to the ideas presented in this paper. Antoine Zimmermann and Andrei Ciortea had funding related to project HyperAgents from grants ANR-19-CE23-0030-01 and SNSF No. 189474. María Poveda-Villalón received funding from the Spanish project KnowledgeSpaces (PID2020-118274RB-I00).

References

1. Berners-Lee, T.: Linked data. Published online by the author as a Web design issue (2006), http://www.w3.org/DesignIssues/LinkedData.html
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5), 34–43 (May 2001), https://www.scientificamerican.com/article/the-semantic-web/
3. Brooks, Rodney, A.: Intelligence Without Reason. In: Mylopoulos, J., Reiter, R. (eds.) Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991. pp. 569–595. Morgan Kaufmann (1991), http://ijcai.org/Proceedings/91-1/Papers/089.pdf
4. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal on Robotics and Automation **2**(1), 14–23 (1986), https://doi.org/10.1109/JRA.1986.1087032
5. Capadisli, S., Berners-Lee, T., Verborgh, R., Kjernsmo, K.: Solid protocol. W3c solid community group working draft, World Wide Web Consortium (Dec 17 2021), https://solidproject.org/TR/2021/protocol-20211217
6. Franklin, S., Graesser, A.: Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In: Müller, J.P., Wooldridge, M.J., Jennings, N.R. (eds.) Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1193, pp. 21–35. Springer (1997), https://doi.org/10.1007/BFb0013570
7. Georgeff, M.P., Lansky, A.L.: Reactive Reasoning and Planning. In: Forbus, K.D., Shrobe, H.E. (eds.) Proceedings of the 6th National Conference on Artificial Intelligence. Seattle, WA, USA, July 1987. pp. 677–682. Morgan Kaufmann (1987), http://www.aaai.org/Library/AAAI/1987/aaai87-121.php
8. Hendler, J.: Where are all the intelligent agents? IEEE Intelligent Systems **22**(03), 2–3 (2007), https://doi.org/10.1109/MIS.2007.62
9. Kaebisch, S., Kamiya, T., McCool, M., Charpenay, V., Kovatsch, M.: Web of Things (WoT) Thing Description. W3C Recommendation, World Wide Web Consortium (Apr 9 2020), http://www.w3.org/TR/2020/REC-wot-thing-description-20200409/
10. Lanthaler, M., Gütl, C.: Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In: Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M., Auer, S. (eds.) Proceedings of the WWW2013 Workshop on Linked Data on the Web, Rio de Janeiro, Brazil, 14 May, 2013. CEUR Workshop Proceedings, vol. 996. Sun SITE Central Europe (CEUR) (May 2013), http://ceur-ws.org/Vol-996/papers/ldow2013-paper-03.pdf
11. Maes, P.: Modeling Adaptive Autonomous Agents. Artificial Life **1**(1_2), 135–162 (10 1993), https://doi.org/10.1162/artl.1993.1.1_2.135
12. Nwana, H.S., Lee, L.C., Jennings, N.R.: Co-ordination in Multi-Agent Systems. In: Nwana, H.S., Azarmi, N. (eds.) Software Agents and Soft Computing: Towards Enhancing Machine Intelligence, Concepts and Applications. Lecture Notes in Computer Science, vol. 1198, pp. 42–58. Springer (1997), https://doi.org/10.1007/3-540-62560-7_37
13. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems **17**(3), 432–456 (Dec 2008), https://doi.org/10.1007/s10458-008-9053-x

14. Sambra, A., Corlosquet, S.: WebID 1.0 - Web Identity and Discovery. W3C IG Editor's draft, World Wide Web Consortium (Apr 30 2015), https://dvcs.w3.org/hg/WebID/raw-file/tip/spec/identity-respec.html
15. Sichman, J.S.a., Conte, R., Castelfranchi, C., Demazeau, Y.: A social reasoning mechanism based on dependence networks. In: Proceedings of the 11th European Conference on Artificial Intelligence. p. 188–192. ECAI'94, John Wiley & Sons, Inc., USA (1994)
16. Speicher, S., Arwe, J., Malhotra, A.: Linked Data Platform 1.0. W3C Recommendation, World Wide Web Consortium (Feb 26 2015), http://www.w3.org/TR/2015/REC-ldp-20150226/
17. Wooldridge, M.: Intelligent Agents. In: Weiss, G. (ed.) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 27–72. The MIT Press (2000), http://mitp-content-server.mit.edu:18180/books/content/sectbyfn?collid=books_pres_0&id=4791&fn=9780262731317_sch_0001.pdf