Principled Design of Web-based Multi-Agent Systems

Andrei Ciortea
University of St.Gallen, Switzerland
andrei.ciortea@unisg.ch

There has been significant work on engineering Web-based multi-agent systems (MAS), but to derive a principled way forward, we need the concerted effort of experts in several areas, such as *Web Architecture and the Web of Things*, *Semantic Web and Linked Data*, and *Autonomous Agents and Multi-Agent Systems*. The WebAgents Community Group provides the ideal platform for fostering such discussions. In what follows, I outline a few ideas I find important at the intersection of these areas — and why I think they should be part of the discussions.

This position statement was initially submitted to the Dagstuhl Seminar 23081 on "Agents on the Web".[1]

Environment as a First-class Abstraction. In the classical AI view, the *environment* is simply the world external to the agent [13]: the agent perceives and acts upon its environment to achieve its design objectives. This view works well for single-agent systems, but in a multi-agent system, multiple agents interact in a shared environment—and thus require a supporting infrastructure to discover other agents, to exchange messages, to coordinate, etc. In practice, such responsibilities have to be ascribed to components within the system, which led to the view of the *environment as a first-class abstraction* in MAS [16]: a component that is designed with clear-cut responsibilities and provides agents with various levels of support (access to their deployment context, interaction, mediation, etc.).[2] Going further, the environment can serve not only as a design abstraction but also as a *programming abstraction* [12]: it can provide a reflective level of support that allows agents to shape their environment such that it better fits their needs.

I think this separation of concerns between agents and the environment abstraction can be fundamental for engineering Web-based MAS. First, it simplifies the architecture of the overall system—and it allows to transpose architectures for MAS to the Web Architecture. Second, it enables evolvability: it allows agents and components in their environment to be developed, deployed, and to evolve independently from one another. Third, it ensures forward compatibility in long-lived Web-based MAS: the environment abstraction can reify other existing or future conceptual dimensions in the MAS, an argument best illustrated by the JaCaMo meta-model and platform[3] [3]. Together with colleagues, we have been developing the idea of hypermedia-based environments as a first-class abstraction in Web-based MAS—and as a conceptual bridge between MAS and the Web [4–7].

Asynchronous Interaction. It is widely accepted that synchronous interactions are insufficient for engineering MAS (e.g., see [14]). The Web was originally built on synchronous interactions between components—but since then, a plethora of methods and protocols have been developed to support asynchronous interaction on the Web: polling, long-polling, webhooks, CoAP Observe, W3C WebSub, WebRTC, etc. Such extensions address requirements beyond the classic Web Architecture, but how such extensions fit within a REST-style Web is not always obvious.

The reason is that REST, as a coherent closed set of architectural design decisions, was defined in the mid-90s to meet the needs of the Web at that time—and asynchronous interaction was not among them [8]. Several generations of researchers guided by Richard Taylor have continued to extend the insights of REST to meet requirements beyond the classic Web Architecture [9]. For example, one of these extensions is *Asynchronous REST*, developed as part of ARRESTED [10], which allows clients to observe Web resources on a server. A similar method is implemented by CoAP via the `Observe` flag [15].[4]

Uniform Interaction. Two ideas that shaped the Web are Sir Tim Berners-Lee's original idea of unifying heterogeneous information systems behind a *simple*, *uniform hypertext interface*—one that all users could access easily and from anywhere [2], and Roy Fielding's idea of a *uniform interface*—one that would allow creating Unix-like pipelines for processing the flow of information on the Web [8]. Both ideas emphasize uniform interaction.

On the Web, uniform interaction is based on a small set of methods with well-defined semantics, and an evolving set of media types that capture the semantics of representations of resources. The semantics of methods and media types are (ideally) agreeable among origin clients[5], origin servers, and any intermediaries in between. In this context, "semantics" simply means consensus among people that is standardized and hard-coded into components. In the Semantic Web view,

[1] https://www.dagstuhl.de/23081

[2] See also the *Environment for MAS* workshop series.

[3] In JaCaMo, the organization dimension is reified through the enviornment. [3]

[4] The Constrained Application Protocol (CoAP) is a Web protocol for constrained devices and networks developed in the context of the Web of Things [15].

[5] In the Web community, what I call "origin clients" are known as "user agents", but I avoid this term for the purpose of this seminar. I use "origin clients" instead because these are the components that originate requests (similar to how origin servers are the components that originate responses).

resource representations use media types for RDF serialization formats and carry logic-based semantics. If resources also link to one another (cf. the Linked Data principles[6]), this then creates a hypermedia-driven, uniformly-accessible, knowledge-level abstraction of the Web.

If we consider the environment as a first-class abstraction in Web-based MAS, where the environment can also reify all the other conceptual dimensions in the MAS, then we have all the elements we need to provide agents with a knowledge-level abstraction of the system that they can interact with [4]. Similar to how we interact through and with the Web, agents can then interact with one another and with the rest of their system (artifacts, organizations, etc.) through the environment dimension by producing and consuming semantic hypermedia. What remains is to ensure that agents are able to both perceive and act on their hypermedia-based environment in a uniform way (i.e., to ensure both *uniform action* and *uniform perception*).

Situatedness and Embodiement. Two notions of particular importance in research on intelligent agents are *situatedness* and *embodiment*: the dominant view is that intelligent behavior 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. In hypermedia-based environments such as the Web, there is no implicit support for situatedness and embodiment: multiple agents can browse the same website but without being able to effect change and without being aware of other agents browsing the website. However, there are also no inherent limitations: if we consider the environment as a first-class abstraction in Web-based MAS, the hypermedia-based environment can be designed to provide agents with various levels of support for situatedness and embodiment—it is merely a design choice.

Decentralization. The Web Architecture assumes centralized resources in a decentralized Web [9]: every resource is hosted on an origin server, but the Web itself transcends geographical and organizational boundaries. This is already a great start for engineering Web-based MAS—and one that still raises many challenges (see the Solid project[7]).

Going further, ARRESTED [10] is a variation of REST that was proposed to meet the requirements of distributed and decentralized resources. Other related work includes IFPS [1] and blockchain-based initiatives for designing a peer-to-peer, versioned Web. Although a source of inspiration, such architectural styles seem to go beyond what could seem like a first set of fundamental building blocks required to engineer Web-based MAS.

[6] https://www.w3.org/DesignIssues/LinkedData.html
[7] https://solidproject.org/

References

[1] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. arXiv:1407.3561 http://arxiv.org/abs/1407.3561

[2] Tim Berners-Lee. 1989. *Information management: A proposal.* Technical Report. CERN. Report No. CERN-DD-89-001-OC.

[3] Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, and Andrea Santi. 2013. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming* 78, 6 (2013), 747 – 761. https://doi.org/10.1016/j.scico.2011.10.004 Special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.

[4] Andrei Ciortea, Olivier Boissier, and Alessandro Ricci. 2019. Engineering World-Wide Multi-Agent Systems with Hypermedia. In *Engineering Multi-Agent Systems*, Danny Weyns, Viviana Mascardi, and Alessandro Ricci (Eds.). Springer International Publishing, Cham, 285–301.

[5] Andrei Ciortea, Olivier Boissier, Antoine Zimmermann, and Adina Magda Florea. 2017. Give Agents Some REST: A Resource-Oriented Abstraction Layer for Internet-Scale Agent Environments. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (São Paulo, Brazil) *(AAMAS '17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1502–1504.

[6] Andrei Ciortea, Simon Mayer, Fabien Gandon, Olivier Boissier, Alessandro Ricci, and Antoine Zimmermann. 2019. A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World Wide Web. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1659–1663.

[7] Andrei Ciortea, Antoine Zimmermann, Olivier Boissier, and Adina Magda Florea. 2016. Hypermedia-Driven Socio-Technical Networks for Goal-Driven Discovery in the Web of Things. In *Proceedings of the Seventh International Workshop on the Web of Things* (Stuttgart, Germany) *(WoT '16)*. Association for Computing Machinery, New York, NY, USA, 25–30. https://doi.org/10.1145/3017995.3018001

[8] Roy Thomas Fielding. 2000. *Architectural styles and the design of network-based software architectures.* Ph. D. Dissertation. University of California, Irvine.

[9] Roy T. Fielding, Richard N. Taylor, Justin R. Erenkrantz, Michael M. Gorlick, Jim Whitehead, Rohit Khare, and Peyman Oreizy. 2017. Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture" (Impact Paper Award). In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (Paderborn, Germany) *(ESEC/FSE 2017)*. Association for Computing Machinery, New York, NY, USA, 4–14. https://doi.org/10.1145/3106237.3121282

[10] R. Khare and R.N. Taylor. 2004. Extending the Representational State Transfer (REST) architectural style for decentralized systems. In *Proceedings. 26th International Conference on Software Engineering*. 428–437. https://doi.org/10.1109/ICSE.2004.1317465

[11] Pattie Maes. 1993. Modeling Adaptive Autonomous Agents. *Artificial Life* 1, 1_2 (10 1993), 135–162. https://doi.org/10.1162/artl.1993.1.1_2.135 arXiv:https://direct.mit.edu/artl/article-pdf/1/1_2/135/1661707/artl.1993.1.1_2.135.pdf

[12] Alessandro Ricci, Michele Piunti, and Mirko Viroli. 2011. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems* 23, 2 (2011), 158–192.

[13] Stuart Russell and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach, Fourth Edition.* Pearson.

[14] Onn M Shehory. 1998. *Architectural properties of multi-agent systems.*

[15] Zach Shelby, Klaus Hartke, and Carsten Bormann. 2014. The Constrained Application Protocol (CoAP). RFC 7252 (Proposed Standard).

http://www.ietf.org/rfc/rfc7252.txt

[16] Danny Weyns, Andrea Omicini, and James Odell. 2007. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems* 14, 1 (01 Feb 2007), 5–30. https://doi.org/10.1007/s10458-006-0012-0

[17] Michael Wooldridge. 2000. Intelligent agents. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence (Second Edition)*, Gerhard Weiss (Ed.). MIT press.