Report from Dagstuhl Seminar 21072

Autonomous Agents on the Web

Edited by

Olivier Boissier[1], Andrei Ciortea[2], Andreas Harth[3], and Alessandro Ricci[4]

1 **Ecole des Mines – St. Etienne, FR,** olivier.boissier@emse.fr
2 **Universität St. Gallen, CH,** andrei.ciortea@unisg.ch
3 **Fraunhofer IIS – Nürnberg, DE,** andreas@harth.org
4 **Università di Bologna, IT,** a.ricci@unibo.it

Abstract

The World Wide Web has emerged as the middleware of choice for most distributed systems. Recent standardization efforts for the Web of Things and Linked Data are now turning hypermedia into a homogeneous information fabric that interconnects everything – devices, information resources, abstract concepts, etc. The latest standards allow clients not only to *browse* and *query*, but also to *observe* and *act on* this hypermedia fabric. Researchers and practitioners are already looking for means to build more sophisticated clients able to meet their design objectives through *flexible autonomous use* of this hypermedia fabric. Such autonomous agents have been studied to large extent in research on distributed artificial intelligence and, in particular, multi-agent systems. These recent developments thus motivate the need for a broader perspective that can only be achieved through a concerted effort of the research communities on *the Web Architecture and the Web of Things*, *Semantic Web and Linked Data*, and *Autonomous Agents and Multi-Agent Systems*. The Dagstuhl Seminar 21072 on "Autonomous Agents on the Web" brought together leading scholars and practitioners across these research areas in order to support the transfer of knowledge and results – and to discuss new opportunities for research on Web-based autonomous systems. This report documents the seminar's program and outcomes.

Seminar February 14–19, 2021 – http://www.dagstuhl.de/21072
2012 ACM Subject Classification Information systems → World Wide Web; Computing methodologies → Multi-agent systems; Human-centered computing → Ubiquitous computing
Keywords and phrases Web Architecture, Web of Things, Semantic Web, Linked Data, Multi-Agent Systems
Digital Object Identifier 10.4230/DagRep.11.1.24
Edited in cooperation with Vachtsevanou, Danai

Autonomous Agents on the Web, *Dagstuhl Reports*, Vol. 11, Issue 01, pp. 24–100
Editors: Olivier Boissier, Andrei Ciortea, Andreas Harth, and Alessandro Ricci
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Olivier Boissier (Ecole des Mines – St. Etienne, FR)
Andrei Ciortea (Universität St. Gallen, CH)
Andreas Harth (Fraunhofer IIS – Nürnberg, DE)
Alessandro Ricci (Università di Bologna, IT)

The vision of autonomous agents on the Web is almost as old as the Web itself: in his keynote at WWW'94[1], Sir Tim Berners-Lee was noting that documents on the Web describe real objects and relationships among them, and if the semantics of these objects are represented explicitly then machines can *browse through* and *manipulate* reality.[2] These ideas were published under the Semantic Web vision in 2001 [1]. Yet in 2007, after having spent the better half of a decade advancing this vision, James Hendler was looking back to conclude that most ideas in the original article were already seeing widespread deployment on the Web except for agent-based systems – and raised the question: "where are all the intelligent agents?" [2].

This question is yet to be addressed. On today's Web, we as *human agents* are often assisted by invisible software agents, such as crawlers used by search engines to navigate and index Web pages, agents that curate online content produced by people (e.g., Wikipedia's content agents), and recommender systems used all over the Web to generate more links and navigation paths (e.g., suggestions of related Web pages). In our everyday lives, we are assisted by more visible agents, such as Amazon's Alexa, Google Duplex, or Apple's Siri. Some of these agents may already use various AI methods (learning, reasoning, etc.), but they are specialized for narrow tasks and constrained to silos dictated by company ecosystems. We have yet to see more autonomous, cooperative, and long-lived agents on the Web [3] – the intelligent agents in James Hendler's question. We believe this decade-old question is now more relevant than ever before in the context of recent developments in three areas of research: (i) *Web Architecture and the Web of Things*, (ii) *Semantic Web and Linked Data*, and (iii) *Autonomous Agents and Multi-Agent Systems*.

The primary objective of this 5-day seminar was to create a network of senior and young researchers who can revisit and align the relevant research threads across the three targeted areas. The seminar was a blend of invited talks, live demonstrators, and group work. Some of the overarching research questions discussed during the seminar included (not an exhaustive list): How to design software agents able to achieve their tasks through flexible autonomous use of hypermedia? How to design hypermedia-based environments that support autonomous behavior? How to design, represent, and reason about interactions among autonomous agents, people, and any other resources on the Web? How to design and govern communities of autonomous agents and people on the Web?

A number of follow-up steps were already taken to continue the discussions and to further consolidate the community. Most recently, several participants submitted a challenge proposal that was accepted at the *20th International Semantic Web Conference (ISWC 2021)*: the *All The Agents Challenge (ATAC)* will take place in October 2021.[3] In addition, a one-day

[1] The First International Conference on the World-Wide Web, CERN, 25-27 May 1994.
[2] Sir Tim Berners-Lee, *The Future of the Web*, WWW'94: `https://videos.cern.ch/record/2671957`, accessed: 07.05.2021.
[3] `https://purl.org/atac/2021`

Figure 1 The traditional end-of-seminar photo on the stairs of Virtual Schloss Dagstuhl.

follow-up event was scheduled for July 9, 2021, and several participants offered to contribute to a shared "live" demonstrator space that would allow to integrate technologies and to try out new ideas across the targeted research areas (see also the working group report in Section 6.1).

The Virtual Seminar Format

The seminar was organized in a fully virtual format due to the COVID-19 pandemic and brought together 45 participants across 4 different time zones. The seminar's schedule was designed for focused online interaction and to preserve to some extent the social dimension specific to regular Dagstuhl Seminars. The schedule was also designed to accommodate the participant's time zones as much as possible. To this end, the schedule was structured around three types of sessions:

- *plenary sessions*: time-zone friendly sessions that constructed the backbone of the seminar (4h per day);
- *Demos & Tech sessions*: sessions reserved for presenting demonstrators and technologies relevant to the seminar (3 sessions of 2h); the objective of these sessions was to ground the discussions and to paint a picture of what can already be achieved with existing technologies;
- *social events*: sessions reserved for online social interactions in a virtual representation of Schloss Dagstuhl via Gather.town[4] (see Figure 1).

The seminar started with five invited talks given by James Hendler, Mike Amundsen, Matthias Kovatsch and Simon Mayer (joint talk), Olivier Boissier, and Dave Raggett (in order of presentation). The invited talks were meant to help bootstrap the discussions and presented developments across the three research areas targeted by the seminar. The reduced virtual format did not allow for additional talks from participants, but we organized several rounds of personal introductions during the first two seminar days.

[4] `https://gather.town/`, accessed: 14.05.2021.

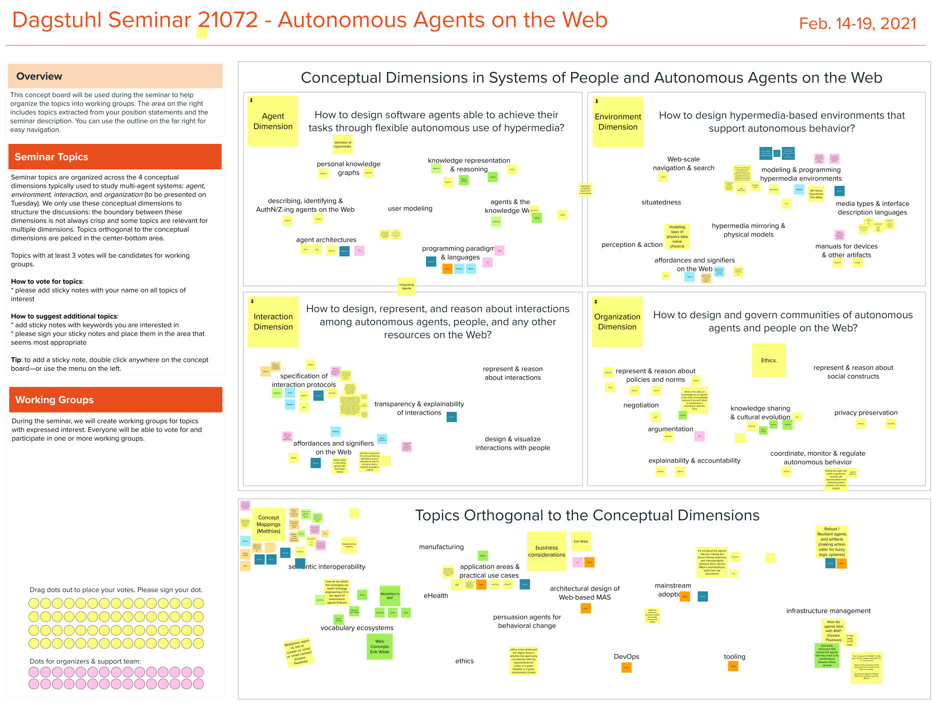


Figure 2 Virtual concept board used to organize the seminar topics into working groups.

In the third seminar day, participants used a virtual concept board (see Figure 2) to organize the seminar topics and to assign them into working groups for the rest of the week. The concept board was created from the seminar topics proposed by the co-organizers, the position statements submitted by participants prior to the seminar, and topics that emerged during the first two seminar days. In total, five working groups were created and four working groups submitted consolidated discussion summaries for this report (see Section 6).

The Demos & Tech sessions attracted more submissions than initially foreseen: we initially scheduled two sessions in the second and third seminar days and eventually scheduled a third session in the fourth seminar day to accommodate all submissions. Submissions were in the form of short abstracts (see Section 5) and each participant was given 10 minutes to present a live demonstrator and 5 minutes for questions.

Overview of the Report

This report is organized into four main parts. Section 3 includes the list of abstracts of the invited talks. Section 4 includes position statements submitted by participants before and after the seminar. Section 5 includes the list of abstracts for the demonstrators presented at the seminar. Section 6 includes the reports submitted by the working groups created during the seminar.

Acknowledgements

The co-organizers would like to thank Fabien Gandon and Simon Mayer for their enthusiasm and support in organizing this seminar. We would also like to thank the team of young researchers who volunteered to support this virtual format: Danai Vachtsevanou (seminar collector), Daniel Schraudner (architect of Virtual Schloss Dagstuhl), Cleber Amaral, Samuele Burattini, Angelo Croatti, Timotheus Kampik, and Adnane Mansour. Finally, we would like to thank Sussanne Bach-Bernhard, Sascha Daeges, and Michael Gerke for their assistance in organizing this seminar in a fully virtual format.

References

1 T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

2 J. Hendler. Where are all the intelligent agents? *IEEE Intelligent Systems*, 22(3):2–3, May 2007.

3 J. Hendler and A. Mulvehill. *Social Machines: The Coming Collision of Artificial Intelligence, Social Networking, and Humanity.* Apress, Berkely, CA, USA, 1st edition, 2016.

2 Table of Contents

3 Overview of Invited Talks

3.1 Where Are All the Intelligent Agents?

James A. Hendler (Rensselaer Polytechnic Institute – Troy, US)

The goal of this talk is to provide some thoughts and challenge problems to help frame the vision of this workshop. From the early agent languages of the mid-1980 to the "agent markup languages" that formed the basis of the semantic web, the agents research community has repeatedly put forth visions of what the agents can do. In this talk, I dredge up some of those old visions and ask why it is, after more than thirty years, we still cannot do many of the things we have long promised.

3.2 From Steve Austin to Peter Norvig : Engineering AMEE, the Simple Autonomous Agent

Mike Amundsen (MuleSoft LLC – San Francisco, US)

Main reference Mike Amundsen: "Autonomous Maze Environment Explorer" (A presentation for Schloss Dagstuhl Seminar [21072] "Autonomous Agents on the Web" – February, 2021). figshare.
URL https://doi.org/10.6084/m9.figshare.14126033.v1

"We can build it. We have the technology." – Oscar Goldman from The Six Million Dollar Man, 1974

Creating and maintaining a simple autonomous agent involves building an application that can successfully deal with Russell & Norvig's (1995) four elements of intelligent agents: Precepts, Actions, Goals, and Environment. Taking a lead from the 1970s US television series "Six Million Dollar Man" we'll explore the details of designing and building AMEE the Agent for Maze Environment Exploration.

Along the way we'll see how you can use existing web technologies such as hypermedia, semantic profiles, and a compact algorithm to build a small Javascript-based autonomous agent that can successfully navigate random two-dimensional mazes of arbitrary size. Finally, we'll contemplate how we can apply what we learned in this example to the larger World Wide Web to create more useful and more sophisticated autonomous agents using existing tools and technologies.

3.3 Human-like AI and the Sentient Web

Dave Raggett (W3C – United Kingdom, GB)

URL https://www.w3.org/2021/sentient-web-2021-01-11.pdf

The seminar organisers cite Jim Hendler, who back in 2007 noted that the Semantic Web still lacked agent-based systems as the basis for services. This remains the case even now in 2021.

Semantic networks and the Aristotelian tradition of mathematical logic have been around for a very long time, but have yet to be widely exploited, despite their popularity in academic circles. Continuing down that path leads to the wasteland of lost dreams and failed ambitions.

Instead, we need to take a fresh look at the natural world with over 500 million years of neural evolution and decades of progress across the cognitive sciences. Human-like AI will enable autonomous agents that think, feel and learn like we do, and are knowledgeable, general purpose, creative, collaborative, empathic, sociable and trustworthy.

Human-like AI can be realised using functional models of the human brain that can be implemented on conventional computer hardware. Human memory can be modelled in terms of the combination of symbolic graphs and sub-symbolic statistical information. Cognition can be modelled in terms of sequential execution of rules together with parallel execution of graph algorithms, along with a blackboard for integrating different systems.

The chunks data and rules language is inspired by John Anderson's work on ACT-R. A chunk represents a collection of properties whose values are literals or references to other chunks, analogous to n-ary terms in RDF. Chunks correspond to the concurrent firing patterns of a bundle of nerve fibres connecting to a given cortical region. Chunks are also easier to work with than RDF. There is an expanding suite of web-based demos, with incubation taking place in the W3C Cognitive AI Community Group.

Current work is focusing on cognitive natural language processing and how to mimic the ease with which young children learn language. Natural language is key to working around the bottleneck imposed by manual knowledge engineering by enabling agents to learn in the classroom and playground. Further out, work is planned on integrating functional models of the limbic system as a basis for emotional intelligence and social interaction.

Relation to the posed Research Questions (see Executive Summary in Section 1): RQ1 and RQ2 refer to hypermedia, but lack a definition of what that is. One interpretation is as a web of declarative knowledge accessible to people and cognitive agents. This needs to be supplemented by a web of cognitive agents that implement value chains in commercially viable ecosystems. RQ3 and RQ4 talk about communities of people, agents and other resources.

My paper and slides describe solutions involving the Sentient Web, the Digital Self, pull-based business models for the future of web search and e-commerce, the end of the culture of digital surveillance, and the emergence of the Web 'verse as distributed AR/VR with avatars for people and cognitive agents, as a quantum leap beyond today's tired Zoom meetings for remote interaction.

Human-like AI will change the world, akin to Alice following the white rabbit down the rabbit hole. Ignore it if you like, as change won't happen immediately, but be aware that huge changes are on their way! Human-like AI will catalyse changes in how we live and work, with human-machine collaboration to boost productivity as population levels decline to a sustainable level. This will involve re-engineering capitalism for the post-industrial era.

What role will you play!

3.4 From the Internet of Things to Autonomous Systems on the Web

Matthias Kovatsch (Huawei Technologies – München, DE) and Simon Mayer (Universität St. Gallen, CH)

In this talk, we motivate the need for autonomy from the Web of Things perspective. We start by introducing the illustrative term "Web with Things", where physical devices are connected to the Web, usually by using application-specific gateways, and which aims at simplifying the manual engineering of systems that incorporate physical devices. Next, we introduce the notion of a "Web on Things", where in adherence with the Web architecture, physical devices are natively integrated into the Web by embedding a (potentially constrained) Web server directly on the Thing itself. This Web-based "Thin Server Architecture" [1] leads to an unbundling of flexible Thing-provided services and mashup applications on top. On the basis of such highly modular systems, we discuss the large potential for the flexible creation of higher-level service mashups and the possibility to automate the required engineering. This automation is necessary, as in dynamic contexts – both regarding goal-driven reconfiguration and fluctuating service availability (e.g., due to mobility) – it becomes impractical to keep developers in the loop for each adaption [3].

Automated mashuping of Things and other services is indeed possible when providing machine-understandable descriptions of Things, which include their data schemas and associated hypermedia controls to describe not only the data model, but the whole interaction model with a Thing. This was a fundamental motivation for the standardization of the W3C Web of Things Thing Description (TD)[5]. However, the initial release of the TD specification predominantly focuses on device management and simple services, and covers only static metadata to mitigate heterogeneity of IoT devices and services. More complex Thing-provided services with multi-step interactions and state changes during service consumption are not yet covered. Current research on these topics focuses specifically on exploiting the concept of "Hypermedia As The Engine Of Application State" (HATEOAS). It is the central principle of REST to enable software agents to flexibly consume truly RESTful Web services through hypermedia-driven *local guidance*. It must be integrated with goal-driven *global guidance*, which considers the goals of these agents. We conclude the talk with a brief discussion of the complexity of planning in dynamic environments [2], where service interactions induce state changes, and possible ways to mitigate these issues through more economic perception by software agents and through the placement of top-down constraints on the planning problem [4].

Our main messages are:

- IoT devices intrinsically have different requirements than classic Web services – connectivity, resource constraints, mobility, etc.
- IoT applications have to deal with physicality, state changes, and generally highly dynamic environments – this must be covered by semantics as well.
- Hypermedia-driven local guidance needs to be integrated with goal-driven global guidance – signifiers and stigmergy might help.

[5] `https://www.w3.org/TR/wot-thing-description/`

References

1 M. Kovatsch, S. Mayer, and B. Ostermaier. Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things. In Proc. IMIS, pages 751–756, 2012.

2 M. Kovatsch, Y.N. Hassan, and S. Mayer. Practical Semantics for the Internet of Things: Physical States, Device Mashups, and Open Questions. In Proc. IoT, pages 54–61, 2015.

3 S. Mayer, R. Verborgh, M. Kovatsch, and F. Mattern. Smart Configuration of Smart Environments. IEEE T-ASE 13(3), 2016.

4 A. Ciortea, S. Mayer, and F. Michahelles. Repurposing Manufacturing Lines on the Fly with Multi-Agent Systems for the Web of Things. In Proc. AAMAS, pages 813–822, 2018.

3.5 Multi-Agent Oriented Programming as a Means to Bring Agents on the Web

Olivier Boissier (Ecole des Mines – St. Etienne, FR)

Joint work of Rafael H. Bordini, Jomi Fred Hübner, Aleassandro Ricci, Andrei Ciortea

The Web is pervasive, increasingly populated with interconnected data, services, people and things [6]. As stated in [7], the Web is the middleware of choice for most distributed systems, where hypermedia turns into a homogeneous information fabric that interconnects everything — devices, information resources, abstract concepts, etc. Clients are not only able to browse and query, but also to observe and act on this hypermedia fabric, transforming the Web in a sophisticated *social machine* [3]. The next step, which has been awaited for a long time, consists in bringing in this global ecosystem, autonomous agents, entities able to react to events while pro-actively defining goals and directing actions to achieve them. Using this hypermedia fabric in a flexible and autonomous manner, agents will build hybrid communities on the Web [2], where they will help people cope with the growing number of available resources, and achieve increasingly complex collaborative tasks.

Research on Multi-Agent Systems (MAS) has led to the development of several models, languages, and technologies for programming not only agents, but also their interaction, the application environment where they are situated, as well as the organisation in which they participate. Research on those topics moved from Agent-Oriented Programming towards Multi-Agent Oriented Programming (MAOP)[1]. A MAS program is then designed and developed using a structured set of concepts and associated first-class design and programming abstractions that go beyond the concepts normally associated with agents. They also include those related to environment, interaction, organisation. As an example of such an approach, JaCaMo is a platform for MAOP[5]. It is built on top of seamlessly integrated dimensions (i.e. structured sets of concepts and associated execution platforms): for programming BDI agents, their artifact-based environments, and their normative organisations. The key purpose of JaCaMo is to support programmers in exploring the synergy between these dimensions, providing a comprehensive and clear programming model, as well as a corresponding platform for developing and running collective autonomous systems.

In this presentation, our claim is that the MAOP framework could serve as the foundations for bringing agents on the Web [4]. Such an approach proposes programming concepts useful to engineer autonomous software agents, to balance reactive and goal-directed behavior in software agents, to define social agents that are able to interact with other agents, to govern their autonomous behavior while evolving under different organisations, norms and policies. By providing environment concepts as first-class entities, it opens new perspectives on the conceptual integration between MAS and the Web in which the Web becomes visible and uniformly accessible to the agents. It could become a place for stigmergic interactions among agents on the Web, that can be programmed for the agents as it is programmed for the people. Web resources are first class abstractions situated in the agent's environment as well as in the digital environment of people. Finally, the organisation concepts as first class abstraction open perspectives to provide the means to govern such hybrid communities on the Web. The Web being visible and accessible as the shared environment in which agents act, it may become the entry point for monitoring the actions on the resources of both the autonomous agents and the people, for monitoring the interactions in these hybrid communities. Thanks to organisation as first-class abstraction, governance and social enforcement mechanisms, external to the agents, could then allow to control and regulate the autonomous social behavior of agents.

However, even if this is the first step to bring agents in such hybrid communities on the Web, having a world-wide scale, open and long lived eco-system, requires still to consider several challenging issues.

References

1 Olivier Boissier, Rafael H. Bordini, Jomi Fred Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. 78(6):747–761.

2 Fabien Gandon, Michel Buffa, Elena Cabrio, Olivier Corby, Catherine Faron-Zucker, Alain Giboin, Nhan Le Thanh, Isabelle Mirbel, Peter Sander, Andrea Tettamanzi, et al. Challenges in bridging social semantics and formal semantics on the web. In *International Conference on Enterprise Information Systems*, pages 3–15. Springer.

3 James Hendler and Alice M Mulvehill. *Social machines: the coming collision of artificial intelligence, social networking, and humanity.* Apress.

4 Andrei Ciortea, Simon Mayer, Fabien Gandon, Olivier Boissier, Alessandro Ricci, and Antoine Zimmermann. 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*, AAMAS '19, pages 1659–1663. International Foundation for Autonomous Agents and Multiagent Systems.

5 Olivier Boissier, Rafael Hector Bordini, Jomi Fred Hubner, and Alessandro Ricci. *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo*. Intelligent Robotics and Autonomous Agents series. MIT Press.

6 Matthias Kovatsch, Ryuichi Matsukura, Michael Lagally, Toru Kawaguchi, Kunihiko Toumura, and Kazuo Kajimoto. Web of Things (WoT) Architecture, W3C Recommendation 9 April 2020.

7 Olivier Boissier, Andrei Ciortea, Andreas Harth, and Alessandro Ricci. Dagstuhl Seminar on Autonomous Agents on the Web. February 14-19, 2021, Schloss Dagstuhl, Warden, Germany.

4 Overview of Position Statements

4.1 Persuasion Agents on the Web

Jean-Paul Calbimonte (HES-SO Valais – Sierre, CH)

Introduction

Persuasion can be seen as the action of convincing someone to do something. In this work we refer to persuasion targeting behavior change [5], e.g., for inducing positive lifestyle habits, increasing physical activity, adopting a healthy diet, improving adherence to a treatment, handling stress levels, or adopting strategies for coping with isolation.

We argue that agents on the Web [3] have the potential to bring autonomous and personalized orchestration of large scale persuasion strategies, while allowing decentralized and privacy preserving data management. Nevertheless, this potential requires exploring challenges related to the heterogeneity of persuadees, personalization of persuasion techniques, ethical constraints, as well as coordination & negotiation strategies among persuasion participants.

Agents & Persuasion

Persuasion is often needed when people manifest the will to adopt new or different habits, resolutions, lifestyle choices, as well as to follow a certain treatment aiming at improving their health. Although in many cases such people may have the intention to stick to their decision, they often require non-negligible support and guidance in order to attain the desired behavior change. In this context, persuasion strategies are of utmost importance, specially if the social dimension and the personalization aspects are taken into consideration [1, 4].

Agents on the Web provide a unique combination that matches three of the key aspects required for persuasion: autonomy, decentralization and interlinking. In fact, agents allow encapsulating both persuaders and persuadees, including their motivations and objectives, while relying on a decentralized governance in which all participants are interconnected [3, 6].

Indeed, agents allow modelling persuadees' goals, as well as expectations and behaviors, without the need of a central authority dictating how their interactions should be played out. Moreover, Web agents are deployed in an environment that is inherently open to the construction of social spaces. In persuasion, the role of peers and social influence is crucial for achieving effective results, and the creation of persuasion agent communities on the Web may offer a collective behavior change in a number of domains.

Challenges

Considering the potential of Agents on the Web for persuasion in behavior change, we identify a number of relevant challenges:

- Heterogeneous domain knowledge: Web Semantics are a deadly weapon against heterogeneity of agents' knowledge and beliefs. Coherent and mutually understandable knowledge among agents is a fundamental step towards persuasion at scale.
- Modelling participant's goals, preferences, and expectations provides the foundations for agent-driven understanding of the persuadee. However, these elements have a subjective dimension which is often hard to capture adequately.

- Social coordination of agents allows communities of persuadees to increase the efficacy of behavior change, not only of the individual but for the entire group.
- Negotiation in persuasion is key for dynamic revision of behavior change strategies, as well as to find compromises that adapt to changing situations and redefinition of goals.
- The risks of manipulation and mischievous behaviors surround the field of computational persuasion. Ethical concerns regarding the implementation of agent-based persuasion need to be integrated into this research road-map.
- Behavior change inherently requires handling sensitive data, which should be subject to strict privacy constraint, while guaranteeing data control & ownership for the persuadee.

Opportunities

Addressing these challenges, persuasion agents on the Web may open a number of research opportunities:

- Exploiting semantics as means to facilitate agent coordination and negotiation towards common goals in persuasion scenarios [8].
- Design and implementation of rational argumentation mechanisms [7] using semantics as the backbone of persuasion interactions via dialogues, e.g. conversational agents.
- Creation of personal Knowledge Graphs used to describe both intra-agent and inter-agent interactions and behaviors.
- Design of explainable and accountable persuasion agents whose actions can be traced, justified, and contrasted against objectives/constraints.
- Domain-specific deployments on different areas including eHealth, prevention, physical activity, diet, mental health, etc.
- Study of ethical agents on the Web, including risks and prevention mechanisms [2], as well as the implementation of transparency & fairness policies within the persuasion environment.

References

1 Elizabeth Black, Amanda J Coles, and Christopher Hampson. *Planning for Persuasion*. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. 2017, pp. 933–942.
2 David Eichmann. *Ethical web agents*. In: Computer Networks and ISDN Systems 28. 1-2 (1995), pp. 127–136.
3 James Hendler. *Agents and the semantic web*. In: IEEE Intelligent systems 16. 2 (2001), pp. 30–37.
4 Christopher Hidey and Kathleen McKeown. *Persuasive influence detection: The role of argument sequencing*. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. 1. 2018.
5 Anthony Hunter. *Towards a framework for computational persuasion with applications in behaviour change*. In: Argument & Computation 9. 1 (2018), pp. 15–40.
6 Sabrina Kirrane. *Intelligent Software Web Agents: A Gap Analysis*. In: arXiv preprint arXiv:2102. 06607 (2021).
7 Bernard Moulin et al. *Explanation and argumentation capabilities: Towards the creation of more persuasive agents*. In: Artificial Intelligence Review 17. 3 (2002), pp. 169–222.
8 Chris Provis. *Negotiation, persuasion and argument*. In: Argumentation 18. 1 (2004), pp. 95–112.

4.2 The Contribution of Modal Logics to Hyperagent Theory

Victor Charpenay (Mines Saint-Étienne, FR)

The Web is a versatile environment. Its architecture can apply to a broad range of systems, from the technical systems of Industry 4.0 to collaborative editing platforms to Social Linked Data: any large-scale decentralized system can be designed in terms of autonomous Web agents. The Web architecture suggests a unified way for agents of perceiving their environment, acting on it and interacting with each other.

The Multi-Agent Oriented Programming (MAOP) framework, materialized by JaCaMo [1], could serve as the foundations of a theory for autonomous Web agents. The theory would also include Web of Things (WoT) principles, to anchor (software) agents in their physical environment and provide them with a body. The Resource Description Framework (RDF), in line with what it has be designed for, would help integrate WoT with MAOP. Agents would then have a mind, a body and a common language, three essential pieces of a unified theory of autonomous Web agents. To give a name to that new class of agents, the "cyberagent" would have been a good candidate – after all, the ambition is to use *cyberspace* as the information layer of *cyber*-physical systems. The term "hyperagent" has settled instead, though, conveying more distinctly where novelty comes from: it comes from the Web.

MAOP and WoT, with RDF in the middle, do not make for a unified theory alone, though. While RDF is a useful abstraction to expose static knowledge of the physical world to agents, it lacks two features that are paramount in multi-agent systems: the relativity of that knowledge (1) to a speaker and (2) to time.

For instance, the architecture of the Web does not prevent different origin servers from exposing inconsistent knowledge. Agents should be able to distinguish between the various servers and trace the provenance of the knowledge they are exposed. RDF also shows some limitations in formalizing the concept of affordance, defined in WoT. Affordances are what allows sensors and actuators to "afford" potential actions to agents. Formally, affordances are defined in RDF only as plain resources providing templates (or "forms") to submit data to Web servers. How should hyperagents select an affordance, among those they find in their environment? How can agents evaluate the relevance of an affordance against its own intentions? The answer lies in the ability of agents to formally entail what consequences submitting a form has on the physical world, a form of temporal reasoning.

The key to the above research questions, in both cases, is to introduce modalities to RDF. To make RDF statements relative to a speaker, use epistemic modal logics. To make them relative to time, use temporal logics. However, applying epistemic or temporal modalities to RDF is an arduous task. The logical language of RDF is OWL, the Web Ontology Language. OWL has been mostly designed for the purpose of semantically aligning various sources of information under an abstract model made of classes and properties. Popular temporal logics such as Linear Temporal Logic or Allen's interval calculus [2] cannot be embedded in the most expressive variant of OWL, based on the $\mathcal{SROIQ}$ Description Logic. OWL is not much more helpful in modeling subjectivity, as it makes no distinction between knowledge and belief.

Early on, the multi-agent community has identified modal logics as suitable foundations to distinguish present events from future ones, potential actions from occurring ones and knowledge from belief. These distinctions are necessary for agents to make observations, commit to choices or influence each other. "Reasoning, after all, is just one form of information handling by rational agents" [3]. By slightly reducing the expressive power of OWL, other forms of information handling can be formalized through epistemic or temporal but also dynamic and deontic modalities. Most importantly, modal logics would help formalize further the concept of affordance defined in WoT, to make it useful to hyperagents.

The premises of a modal language have recently been introduced with RDF-star[6] but the language does not include traditional logical connectors such as negation, disjunction and implication. Introducing modalities to RDF is an important step forward in theorizing hyperagents and much remains to be done in that respect.

References

1 O. Boissier et al. (2020), *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo*, Intelligent Robotics and Autonomous Agents Series, MIT Press.
2 M. Fisher et al. (2005), *Handbook of Temporal Reasoning in Artificial Intelligence*, Foundations of Artificial Intelligence Series, Elsevier.
3 J. van Benthem (2010), *Modal Logic for Open Minds*, CSLI Lecture Notes, Center for the Study of Language and Information.

4.3 The Web as a Culture Broth for Agents and People to Grow Knowledge

Jérôme Euzenat (INRIA – Grenoble, FR)

The worldwide web is, in particular, a fantastic medium through which people and organisations share knowledge. This initial ambition of the web has not faded, though the web is used for many other purposes and efforts are deployed to privatise part of it.

Agents on the web

There are plenty of agents on the Web already, they behave like tiny machines: as consumers, harvesting knowledge through crawling the web, as builders, developing the web through connecting and adding knowledge, as repairers, finding dangling links, contradicting statements, and fixing then. These are some basic tasks by which such agents can contribute to the web. And they do it, for instance as wikipedia bots.

As long as agents are autonomous, some of them will unweave what others woven. This is also what people do. But the web will be more solid as agents were to share a weaving culture. How can this be when they are autonomous? Agents have to better communicate with each others. They can do this in various ways: direct communication, cooperation or taking the web as a transportation layer, but also as a medium. Thanks to the web, agents can be autonomous and social, not autonomous and lonely.

Agents as knowledge producers

Like yeast refines sugar into alcohol, agents may refine web data into knowledge and beliefs. They may perform such tasks at a relatively basic level based on syntactic or statistical hints. They may do it as well by developing a deeper understanding of the web content and the knowledge that underlies it.

---

[6] `https://w3c.github.io/rdf-star/cg-spec/`

The semantic web effort [1] was one step in this direction. The ability to express formal knowledge, as ontologies, and to understand data through this knowledge is a stepping stone on which agents could build. It has led to proposals such as a data washing machine built on web knowledge itself [2].

Knowledge as culture

Knowledge and beliefs are socially elaborated as part of the culture of a society. Sharing knowledge is building cultures, different cultures. We should study how agents can elaborate knowledge socially and culturally [3]. This falls into the topic of cultural evolution, in which evolution theory is applied to such phenomena [4]. We should study how they can evolve their knowledge throuh wandering, learning, building or repairing the web.

There may be many ways to study this: analysing how human societies do it, developing logical theories of knowledge evolution, or experimenting with cultural knowledge evolution. We are currently running a program [5, 6], building on the work on cultural language evolution [7], to deepen our understanding of how agents may evolve their knowledge.

But, like people, it would be nice that agents have a life beyond the web.

Agents and people

Human beings are in the web like fishes in the sea. The web, as a mediation architecture, has been massively adopted, in various modalities by humans beings. Yet, it is a fully artificial environment, so eventually prone to be adopted by agents.

As discussed in previous sections, the building of a culture is only achieved through interaction, either implicit or explicit, and adapting behaviour with respect to interaction. Because the web could be an environment natural to both agents and human beings, it is a perfect playground for studying further cultural evolution in agents driven by human-compatibility [8]. Actually, it may also be a good playground to study as well how human-compatible are some human behaviour.

References

1. Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, *Scientific american* 284(5):34–43, 2001.
2. Sören Auer, Jens Lehmann, Creating knowledge out of interlinked data, *Semantic web journal* 1(1–2):97–104, 2010.
3. Jérôme Euzenat, A map without a legend: the semantic web and knowledge evolution, *Semantic web journal* 11(1):63–68, 2020.
4. Alex Mesoudi, Alan White, Kevin Laland, Towards a unified science of cultural evolution, *Behavioral and brain sciences* 29(4):329–383, 2006.
5. Jérôme Euzenat, Interaction-based ontology alignment repair with expansion and relaxation, Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI), Melbourne (AU), pp185–191, 2017.
6. Yasser Bourahla, Manuel Atencia, Jérôme Euzenat, Knowledge improvement and diversity under interaction-driven adaptation of learned ontologies, Proc. 20th International conference on autonomous agents and multiagent systems (AAMAS), Online, 2021, to appear.
7. Luc Steels (ed.), Experiments in cultural language evolution, John Benjamins, Amsterdam (NL), 2012.
8. Stuart Russell, Human compatible: artificial intelligence and the problem of control, Viking New-York (NY US), 2019.

4.4 Merry hMAS and Happy New Web: A Wish for Standardizing an AI-Friendly Web Architecture for Hypermedia Multi-Agent Systems

Fabien Gandon (INRIA – Sophia Antipolis, FR)

Main reference Andrei Ciortea, Simon Mayer, Fabien Gandon, Olivier Boissier, Alessandro Ricci, Antoine Zimmermann: "A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World Wide Web", in Proc. of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019, pp. 1659–1663, International Foundation for Autonomous Agents and Multiagent Systems, 2019.
URL http://dl.acm.org/citation.cfm?id=3331893
Main reference Simon Bienz, Andrei Ciortea, Simon Mayer, Fabien Gandon, Olivier Corby: "Escaping the Streetlight Effect: Semantic Hypermedia Search Enhances Autonomous Behavior in the Web of Things", in Proc. of the 9th International Conference on the Internet of Things, IoT 2019, Bilbao, Spain, October 22-25, 2019, pp. 28:1–28:8, ACM, 2019.
URL http://dx.doi.org/10.1145/3365871.3365901
Main reference Fabien Gandon: "The Web We Mix: Benevolent AIs for a Resilient Web", in Proc. of the 11th ACM Conference on Web Science, WebSci 2019, Boston, MA, USA, June 30 – July 03, 2019, pp. 115–116, ACM, 2019.
URL http://dx.doi.org/10.1145/3292522.3329406
Main reference Fabien Gandon: "A survey of the first 20 years of research on semantic Web and linked data", Ingénierie des Systèmes d Inf., Vol. 23(3-4), pp. 11–38, 2018.
URL http://dx.doi.org/10.3166/isi.23.3-4.11-38
Main reference Olivier Corby, Catherine Faron-Zucker, Fabien Gandon: "LDScript: A Linked Data Script Language", in Proc. of the The Semantic Web – ISWC 2017 – 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 10587, pp. 208–224, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-68288-4_13
Main reference Fabien Gandon: "The Three 'W' of the World Wide Web Call for the Three 'M' of a Massively Multidisciplinary Methodology", in Proc. of the Web Information Systems and Technologies – 10th International Conference, WEBIST 2014, Barcelona, Spain, April 3-5, 2014, Revised Selected Papers, Lecture Notes in Business Information Processing, Vol. 226, pp. 3–15, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-319-27030-2_1
Main reference Fabien Gandon, Michel Buffa, Elena Cabrio, Catherine Faron-Zucker, Alain Giboin, Nhan Le Thanh, Isabelle Mirbel, Peter Sander, Andrea G. B. Tettamanzi, Serena Villata: "Challenges in Bridging Social Semantics and Formal Semantics on the Web", CoRR, Vol. abs/1408.7092, 2014.
URL http://arxiv.org/abs/1408.7092

Although it was initially an "Information management proposal" [1] the Web really is a successful « application integration platform » [2]. More importantly, it is both. It is a self-documenting hypermedia system for application and information integration, and that makes the Web very special. Just like it is important to propose a Web of linked data and distributed RDF knowledge graphs as an alternative to data silos [3], the Web must also support an alternative to intelligence silos and thrive to host a wealth of distributed artificial and natural intelligence forming hybrid communities [6] and managing distributed resources [10]. This is a call for hMAS: Hypermedia Multi-Agent Systems [4] [5].

Among the success factors of the Web are a number of non-functional properties purposefully enforced at design time including: simplicity, generality, portability, extensibility and the systematic search for compatibility [8]. For instance, the simplicity meant, at the time, accepting to simplify solutions (e.g. SGML vs. HTML, HTTP vs other protocols) but also proposing viral approaches for adoption to learn to "weave by weaving" including copy-paste facilities of Web pages codes to start to contribute and what will later become the wiki-way too. The fact the Web proposal was put in the public domain by CERN in 1993 was also critical in reaching the threshold in volume of attractive resources and trigger the network effects and Metcalfe's law. All these lessons learnt should be kept in mind when specifying hMAS.

Making autonomous agents a first class abstraction of the Web architecture requires importing important MAS concepts (environment, workspace, platform, situatedness, observability, organizations, norms, regulation, interactions). But to maintain extensibility and forward compatibility hMAS must be a friendly architecture for all types of agents [9]: reactive vs cognitive, knowledge reasoning vs connectionist, stygmergy-oriented vs contractual protocol ones,... In this picture, semantic Web and linked data [7] have a key role to play both in weaving the hypermedia fabric for the agents' environment and in providing the semantics to capture the key concepts of hMAS in ontologies to ensure interoperability. Extensible top ontologies are needed to setup the architecture and support extension by domain-dependent and task-dependent ontologies needed for practical concrete applications.

Linked data also come with solutions and concepts to be used and aligned with hMAS for instance LDP [11] principles and containers or languages such as SPARQL to manipulate RDF data, SCHACL to validate and exchange constraints, and extensions such as LDScript to program on top of linked data [12], a language that could be a candidate to align with agent programming languages. For instance, the notion of norms in MAS could be positioned w.r.t. rule languages and validation languages like SHACL which use cases already include both the validation of outputs and inputs of a software and the validation of interactions with a human like we would have in hybrid communities.

In parallel, Web of things, thin servers [13] and Digital Twins [14] are giving more and more substance to the Web resources that shadow physical resources. The URI could lead to more and more informed Web resources and put in touch a variety of digital twins and autonomous agents with the potential of supporting multi-model approaches at an unprecedented scale. Here again, the linked data framework holds the potential for deeply linking all these models [15].

To conclude, one of the hardest tasks for Tim Berners-Lee in the early 90s was to make people imagine a world with a fully deploy Web. Years later, it is hard to imagine a world without the Web. We have the same cold-start problem with the hMAS and we need to find incentives for this change to happen and to reach the threshold in terms of usage that will trigger the network effect and make it go viral. Therefore, together with a proposal of a standard architecture, we need to find incentives and added values for hMAS to be taken on by industry and developers.

References

1 Tim Beners-Lee. *Information Management: a proposal.* CERN, 1989.

2 Roy Fielding, Richard Taylor, Rohit Khare . *Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture"*, FSE'17 Keynote, Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017), pp. 4-11

3 Mansour, Essam, Andrei Vlad Sambra, Sandro Hawke, Maged Zereba, Sarven Capadisli, Abdurrahman Ghanem, Ashraf Aboulnaga, and Tim Berners-Lee. *A demonstration of the solid platform for social web applications.* In Proceedings of the 25th International Conference Companion on World Wide Web, pp. 223-226. 2016.

4 Ciortea A., Boissier O., and Ricci A., 2019. *Engineering World-Wide Multi-Agent Systems with Hypermedia.* In: Weyns D., Mascardi V., Ricci A. (eds) Engineering Multi-Agent Systems. EMAS 2018. Lecture Notes in Computer Science, vol 11375. Springer, Cham. https://doi.org/10.1007/978-3-030-25693-7_15

5 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 (AAMAS '19). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1659–1663.

6 Gandon, F., Buffa, M., Cabrio, E., Corby, O., Faron-Zucker, C., Giboin, A., Le Thanh, N., Mirbel, I., Sander, P., Tettamanzi, A. and Villata, S., 2013, July. *Challenges in bridging*

social semantics and formal semantics on the web. In International Conference on Enterprise Information Systems (pp. 3-15). Springer, Cham.
7 Fabien Gandon. *A Survey of the First 20 Years of Research on Semantic Web and Linked Data.* Revue des Sciences et Technologies de l'Information – Série ISI : Ingénierie des Systèmes d'Information, Lavoisier, 2018
8 Fabien Gandon. *For everything: Tim Berners-Lee, winner of the 2016 Turing award for having invented... the Web.* 1024 : Bulletin de la Société Informatique de France, Société Informatique de France, 2017, pp.21
9 Fabien Gandon. *Combining reactive and deliberative agents for complete ecosystems in infospheres.* IEEE/WIC International Conference on Intelligent Agent Technology (IAT), Oct 2003, Halifax, Canada.
10 Fabien Gandon, Laurent Berthelot, Rose Dieng-Kuntz. *A multi-agent platform for a corporate semantic web.* AAMAS 2002, 6th International Conference on Autonomous Agents, 5th International Conference on Multi-Agents Systems, 9th International Workshop on Agent Theories Architectures and Languages, Jul 2002, Bologna, Italy.
11 Le Hors, Arnaud J., and Steve Speicher. *The linked data platform (LDP).* Proceedings of the 22nd International Conference on World Wide Web. 2013
12 Olivier Corby, Catherine Faron Zucker, Fabien Gandon. *LDScript: a Linked Data Script Language.* International Semantic Web Conference, Oct 2017, Vienne, Austria.
13 M. Kovatsch, S. Mayer, and B. Ostermaier. *Moving Application Logic from the Firmware to the Cloud:Towards the Thin Server Architecture for the Internet of Things.* InProc. IMIS, pages 751–756, 2012
14 Ricci, A. *A Challenge for "Autonomous Agents on the Web"* &*Friends: Shaping* & *Designing "National Digital Twins".*, University of Bologna, position paper in this seminar, Dagstuhl Seminar, February 2021
15 Fabien Gandon. *From linked data & knowledge graphs to linked intelligence & intelligence graphs or the potential of the semantic Web to break the walls between semantic networks and computational networks*, ISWC 2020 Vision Track Talk https://www.youtube.com/watch?v=b9GPOOu2PTM

4.5 About the Place of Agents in the Web

Jomi Fred Hübner (Federal University of Santa Catarina, BR)

The place of agents in the web is here considered as an integration issue. Both agents and other systems have benefits with the integration, essentially because they are the best technology for differente (and to some extent disjunct) problems. When integrating agents with web entities or resources, we face two cases: (1) agents accessing web entities; and (2) web entities accessing agents. While the former case is addressed by many proposals, the latter case is mostly ignored. Of course, these cases are not viable if the web is considered simply as a kind of transport layer. A proper integration requires a better place for the web.

The most common view for the integration is that the web is the environment for the agents. They access the web resources by perceiving and acting on it. However, for a web application to access an agent, the solution is not so straightforward. Possible solutions are: (i) the application has to provide particular perception for the agent and waits for its action; (ii) the application has to be agentified and so recognised by the agent as a pair it can interact with using an Agent Communication Language. The first solution requires that the agent is perceiving the web application, a quite strong requirement. The second solution requires that we model the web application as an agent, something that could be conceptually inappropriate. In both solutions, the web application has to be modified or adapted for the agent. In this view, the integration exists essentially for the benefit of the agents.

Another view is to transform agents into web resources, they should be resourcefied and thus easily accessed by web applications. The consequences of reducing (cognitive) agents to resources are not clear for me and deserves further investigation. Is it the case of simply providing a common web API for the agents? Anyway, agents will be useful for the web only if they can be easily used by the web.

Briefly, I propose that we should look for an approach where agents and web applications can be integrated as they are, without adaptation in any side.

4.6 The Notion of an Agent as a Practical Software Engineering Abstraction

Timotheus Kampik (University of Umeå, SE)

Joint work of Timotheus Kampik, Cleber Jorge Amaral

In today's interconnected information technology ecosystems, software engineering abstractions that are designed to encapsulate autonomous behavior are of increasing relevance. In the academic Artificial Intelligence community, the notion of an agent is the most prominent concept in this regard, and an active sub-community works on agent-oriented software engineering abstractions, in particular on Agent-Oriented Programming (AOP) languages and frameworks [1]. While existing AOP variants excel at facilitating scientific exploration at the intersection of academic programming and knowledge representation & reasoning, making the case for agent-orientation in practical software engineering is an open problem [2]. To work toward a solution, minimally viable agent-oriented abstractions can be designed in the context of mainstream programming languages and frameworks. As to our understanding, these abstractions should:

- Provide a clear and intuitive value proposition to a (perhaps somewhat intellectually curious) practicing software engineer – even one without a Computer Science degree;
- Not add any non-essential overhead that steepens the learning curve and hampers adoption, for example by imposing languages, dialects, or paradigms that do not fit into modern software development processes and toolchains;
- Come with clear design principles with regard to agent internals (reasoning cycle) and agent interface design (interaction and discoverability).

Potential insights can be gained from the relatively recent spread of functional programming flavors to mainstream and originally predominantly object-oriented programming languages and frameworks, a high-profile example being React.js[7] in the context of JavaScript.

References

1 Shehory, O., and Sturm, A. *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks.* Springer Berlin Heidelberg, 2014.

[7] `https://reactjs.org/`

2 Mascardi, V., Weyns, D., Ricci, A., Earle, C. B., Casals, A., Challenger, M., Chopra, A., Ciortea, A., Dennis, L. A., Díaz, A. F., El Fallah-Seghrouchni, A., Ferrando, A., Fredlund, L.-r., Giunchiglia, E., Guessoum, Z., Günay, A., Hindriks, K., Iglesias, C. A., Logan, B., Kampik, T., Kardas, G., Koeman, V. J., Larsen, J. B., Mayer, S., Méndez, T., Nieves, J. C., Seidita, V., Teze, B. T., Varga, L. Z., and Winikoff, M. Engineering multi-agent systems: State of affairs and the road ahead. *SIGSOFT Softw. Eng. Notes 44*, 1 (Mar. 2019), 18–28.

4.7 Read-Write Linked Data and the Web Architecture as Substrate for Agent-based Information Systems

Tobias Käfer (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Tobias Käfer, Andreas Harth
Main reference Tobias Käfer, Andreas Harth: "Rule-based Programming of User Agents for Linked Data", in Proc. of the Workshop on Linked Data on the Web co-located with The Web Conference 2018, LDOW@WWW 2018, Lyon, France April 23rd, 2018, CEUR Workshop Proceedings, Vol. 2073, CEUR-WS.org, 2018.
URL http://ceur-ws.org/Vol-2073/article-05.pdf
Main reference Tobias Käfer, Andreas Harth: "Specifying, Monitoring, and Executing Workflows in Linked Data Environments", in Proc. of the The Semantic Web – ISWC 2018 – 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 11136, pp. 424–440, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-030-00671-6_25

The web architecture is an architecture that has scaled to a global information exchange infrastructure, but it comes with peculiar constraints, summarised in the REST architectural style [7]. Augmented with knowledge representation using semantic technologies, thus forming Read-Write Linked Data [1], the web presents us with a substrate for integration on the component interaction and data level, i.| e.| a substrate for interoperability. While interaction, data, and component descriptions have been standardised e.g. in [6, 5], a commonly agreed suitable abstraction for describing behaviour has not yet established itself. This may be because there are theoretical foundations for behaviour on Read-Write Linked Data such as [9] have been missing, or established behaviour descriptions from other disciplines such as workflow management require glue and adaption such as [10, 11] before they can be applied on Read-Write Linked Data.

Even once behaviour execution is clear, the sheer number of system components and devices available already today call for assistance for composition and coordination. To this end, more properties of the web architecture such as hypermedia, more standards such as the Thing Descriptions [8], the Linked Data Platform [13], and Linked Data Notifications [3], together with more Artificial Intelligence agent [12] and multi-agent systems [2] techniques need to be intelligently layered on top [4] in order to fully exploit Read-Write Linked Data as architecture to tackle the interoperability challenges of information systems.

References

1 T. Berners-Lee. *Read-Write Linked Data.* Design Issues. 2009. URL: `http://www.w3.org/DesignIssues/ReadWriteLinkedData.html`.
2 O. Boissier, R. H. Bordini, J. F. Hübner and A. Ricci. "Dimensions in programming multi-agent systems". In: *Knowledge Engineering Review* 34 (2019).
3 S. Capadisli and A. Guy. *Linked Data Notifications.* W3C Recommendation. 2017. URL: `https://www.w3.org/TR/ldn/`.

4 A. Ciortea, S. Mayer, F. L. Gandon, O. Boissier, A. Ricci and A. Zimmermann. “A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World Wide Web”. In: International Foundation for Autonomous Agents and Multiagent Systems, 2019.
5 R. Cyganiak, D. Wood and M. Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. 2014. URL: `https://www.w3.org/TR/rdf11-concepts/`.
6 R. Fielding and J. Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. RFC 7230 (Proposed Standard). Internet Engineering Task Force, 2014. URL: `http://www.ietf.org/rfc/rfc7230.txt`.
7 R. Fielding. “Architectural Styles and the Design of Network-based Software Architectures”. PhD thesis. University of California, Irvine (CA), USA, 2000.
8 S. Käbisch, T. Kamiya, M. McCool, V. Charpenay and M. Kovatsch. *Web of Things (WoT) Thing Description*. W3C Recommendation. 2020. URL: `https://www.w3.org/TR/wot-thing-description10/`.
9 T. Käfer and A. Harth. “Rule-based Programming of User Agents for Linked Data”. In: 2018.
10 T. Käfer and A. Harth. “Specifying, Monitoring, and Executing Workflows in Linked Data Environments”. In: 2018.
11 T. Käfer, B. Jochum, N. Aßfalg and L. Nürnberg. “Specifying and Executing User Agents in an Environment of Reasoning and RESTful Systems using the Guard-Stage-Milestone Approach”. In: *Journal on Data Semantics* (). To appear.
12 S. J. Russell and P. Norvig. *Artificial intelligence – a modern approach*.
13 S. Speicher, J. Arwe and A. Malhotra. *–Linked Data Platform 1.0* . W3C Recommendation. 2015. URL: `https://www.w3.org/TR/ldp/`.

4.8 Pervasive Autonomous Systems

Simon Mayer (Universität St. Gallen, CH)

Joint work of Simon Mayer, Andrei Ciortea, Matthias Kovatsch, Danai Vachtsevanou

Based on the decoupling of application logic from the firmware of IoT devices by modeling sensors and actuators of connected devices as Web resources, “Thin Servers” [1] create a rich playground for the construction of device and service mashups [2]. However, to exploit the potential of this playground machine-interpretable descriptions are required not just of device and service interfaces (to enable the automatic execution of mashups), but of device and service *capabilities*. Recently, important first steps towards such descriptions have been concluded by the *World Wide Web Consortium*, in the form of the *W3C WoT Thing Description*.

Looking ahead, the embedding of additional, higher-level, metadata about devices and services that is integrated with the W3C WoT TD and describes artifact capabilities and consequences of their execution [3] is a necessary next step to allow the further automation of device and service integration across domains. Especially within the Web of Things field, it would furthermore be highly beneficial to attempt the integration of such models with numerical (e.g., physics-based) models of real-world environments and artifacts. Inspired by recent work on *Hybrid Systems* [4], this would enable fascinating applications where systems integrate semantic metadata about artifact interfaces and capabilities with (numerical) models of their surroundings and of themselves. Following the *thin server* idea, involved models in a system would then be exposed as individual Web resources and integrated into functional composites, for instance via approaches such as *Linked Open Code* [5]; and such concepts could be taken further into the integration of semantically-tethered clauses in the feature/reward design of reinforcement learning systems.

The integration of semantic, numerical, and interface models across domains leads to high complexity in the service composition process. Already without the integration of numerical models in its reasoning, traditional automated planning approaches fail since "proof[s] must consider the possible environment states which significantly increases the state space for the reasoner, leading to longer execution times" [6]. Automated planning thus requires support, which might be delivered top-down (by constraining the planning problem) and/or bottom-up (during the service selection process). Bottom-up, hypermedia-driven "local guidance" – i.e., REST's HATEOAS principle – might be promising, and it should be explored how carefully designed signifiers [7] could enable local guidance for software agents on top of W3C WoT TDs. Top-down, users could be supplied with an interface to specify constraints around the planning problem and thereby facilitate the planner's task, for instance through the integration of automated planning with multi-agent oriented programming [8].

The integration of concepts, models, and results from the fields of knowledge engineering, mathematical modeling, autonomous agents, and human-computer interaction through the homogeneous information fabric of hypermedia that is proposed in this abstract and advanced with our Dagstuhl Seminar on *Autonomous Agents on the Web* is highly timely, highly promising, and highly stimulating – we need to, together and across communities, identify the challenges on our path to such *ideal* human-machine and machine-machine systems, and define an agenda to overcome them.

References

1 M. Kovatsch, S. Mayer, and B. Ostermaier. Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things. In Proc. IMIS, pages 751–756, 2012.
2 D. Guinard and V. Trifa. Towards the Web of Things: Web Mashups for Embedded Devices. In Adj. Proc. WWW (Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web), pages 1506–1518, 2009.
3 R. Verborgh, T. Steiner, D. Van Deursen, S. Coppens, J. Gabarró Vallés, and R. Van de Walle. Functional Descriptions as the Bridge between Hypermedia APIs and the Semantic Web. In Proc. WS-REST, pages 33–40, 2012.
4 Y.K. Tan and A. Platzer. Switched Systems as Hybrid Programs, 2021. https://arxiv.org/abs/2101.06195.
5 A.E.A. Djebri, A. Ettorre, and J. Mortara. Towardsa Linked Open Code. Under review, accessible at https://openreview.net/forum?id=CEnMNhWGG6P.
6 M. Kovatsch, Y.N. Hassan, and S. Mayer. Practical Semantics for the Internet of Things: Physical States, Device Mashups, and Open Questions. In Proc. IoT, pages 54–61, 2015.
7 D.A. Norman. The Design of Everyday Things. BasicBooks, 2013.
8 A. Ciortea, S. Mayer, and F. Michahelles. Repurposing Manufacturing Lines on the Fly with Multi-agent Systems for the Web of Things. In Proc. AAMAS, pages 813–822, 2018.

4.9 A Challenge for Autonomous Agents on the Web & Friends: Shaping & Designing "National Digital Twins"

Alessandro Ricci (Università di Bologna, IT)

In this position paper, I would like to consider a challenge that may be effective to reason and exchange ideas on different key issues that concern "Autonomous Agents on the Web", as reported in the seminar description. The challenge is about shaping/designing "National Digital Twins" (NDT).

National Digital Twins are about *large-scale open ecosystems of connected digital twins*. A main reference for that is the "National Digital Twin Programme" at CDBB (Centre for Digital Built Britain)[8] based on the "Gemini principles"[9], and other similar efforts in literature [1]. Here I will consider an even broader perspective, besides specific domains.

A Digital Twin (DT) is a digital replica or virtual image of some physical asset, coupled at real-time with its physical twin, available as a service/component on the network, with a proper API. The physical asset can be a physical object/resource, but also a place, a person, or a process which occurs in the physical world. A DT can be used to track, monitor a physical asset, but also to simulate its behaviour in what-if scenarios or forecast what is going to happen, as well as augment its functionalities. A DT follows the entire engineering lifecycle of the physical twin – from design time to operation time. DTs originated in aerospace, more recently have become a mantra for Industry 4.0, and nowadays they are more and more considered in broader contexts (smart cities, healthcare,..) [2].

The "National Digital Twin" view calls for adopting DT as a paradigm for a vision of pervasive digital transformation where DTs are uniformly used to virtualize any strategic physical assets of organisations/institutions, in an open, large-scale and cross-domain perspective. It accounts for open and dynamic ecosystems of connected digital twins, in a system of systems perspective. Such ecosystems become the digital fabric mirroring the physical world – as base OS-like infrastructure to be exploited by independent smart applications running on top. Actually, this strongly recalls the vision of "Mirror World" as injected by David Gelernter three decades ago [3], and more recent extensions [4].

Indeed WoT, (Semantic) Web, and Autonomous Agents appear to be main ingredients for conceiving NDTs. However, shaping infrastructures to design and implement a NDT puts forth some interesting challenges and opportunities about how to put together these ingredients.

Main questions

- *[NDT as hypermedia-based environments / dynamic distributed KG]* – Could NDTs be effectively modelled as hypermedia-based environments? Could Semantic Web models and technologies be effective in modeling both the structure and dynamics (events) of NDTs at the knowledge level?
- *[Cognitive agents observing and reasoning about dynamic distributed KG]* – NDT calls for autonomous agents that are capable of observing and reasoning about (dynamic and distributed) knowledge graphs. How to effectively integrate this capability in cognitive architectures e.g. BDI?

[8] https://www.cdbb.cam.ac.uk/what-we-do/national-digital-twin-programme
[9] https://www.cdbb.cam.ac.uk/DFTG/GeminiPrinciples

- *[agents exploiting simulation]* – DT may be used to simulate future behaviour of the physical asset. This opens the door to think about intelligent agents reasoning not only about the current / past events, but also about future/forecast events/situations, to take decisions.

References

1 Ricci, A., Croatti, A., & Montagna, S. (2021). *Pervasive and Connected Digital Twins–A Vision for Digital Health.* IEEE Internet Computing.

2 Saracco, R. (2019). *Digital twins: Bridging physical space and cyberspace.* Computer, 52(12), 58-64.

3 Gelernter, D. (1993). *Mirror worlds: Or the day software puts the universe in a shoebox... How it will happen and what it will mean.* Oxford University Press.

4 Ricci, A., Piunti, M., Tummolini, L., & Castelfranchi, C. (2015). *The mirror world: Preparing for mixed-reality living.* IEEE Pervasive Computing, 14(2), 60-63.

4.10 Stigmergy and REST – A Perfect Match for Agents on the Web?

Daniel Schraudner (Universität Erlangen-Nürnberg, DE)

Simple beings like ants can show a relatively complex but nonetheless efficient behaviour when working together and sharing relevant information thorugh their environment. Those ants easily can be modelled as simple reflex agents using stigmergy. Most Multi-Agent Systems in contrast however coordinate by sending direct messages from agent to agent. This leads to the need for rather complex agents as they have to maintain an inner model („beliefs“) of the world to take informed decisions. By using the environment as a means of communications agents can share their knowledge with other agents more easily. Model-based reflex agents e. g. could be able to externalize their internal model thus being reduced to simple reflex agents, the simplest form of agents according to [1].

There already have been many approaches to bring Multi-Agent Systems to the Web. Early approaches mainly relied on RPC-style web services like the WS-* standards and SOAP. Also the *Foundation for Intelligent Physical Agents* (FIPA) provided a set of standards to enable autonomous agents to communicate using Web protocols like HTTP.

However, even though using technologies from the Web, all those approaches only used the Web as a communication layer and thus were not aligned to the the architectural style of the application layer of the Web (e. g. FIPA also allows to send messages via SMTP instead of HTTP). Failing to fullfill the architectural constraints of REST, those approaches will not profit from the archtiectural properties of web applications (better scalability, loose coupling, caching, etc.) [2].

Recently there have been approaches following the architectural style of REST more closely. *Hypermedia Multi-Agent Systems* [3] specifically focus on the *Hypermedia as the Engine of Application State* (HATEOAS) constraint to provide agents with a uniform way to discover new agents and artifacts and interact with them. Although this definitely goes in the right direction, we think that the other constraints that comprise REST should not be neglected.

According to Fielding [4], the following contraints apply for RESTful applications:

- Client-Server Architecture
- Statelessness
- Cacheability
- Uniform Interface
- Layered System
- Code-On-Demand (optional)

We propose that by applying the communication paradigm of stigmergy to Multi-Agent Systems in the Web makes it much easier to fulfill the architectural constraints of REST:

4.10.1 Client-Server Architecture

Allowing agents to only directly communicate with artifacts brings a clear seperation between clients (agents) and servers (artifacts). Clients can manipulate the state of the servers but not of each other (only indirectly by writting and reading something to resp. from a server).

4.10.2 Statelessness

Stateless communication between agent and artifact is much easier to achieve than between agent and agent. Each artifact has a distinct resource state that is shared among all agents and thus communication with the artifact can be stateless. Communication between two agents in contrast will depend on the current context of the agents.

4.10.3 Cacheability

Agent-Artifact communciation is cacheable more easily because the transferred message will always be a representation of an artifact state. Among agents on the contrary, arbitrary message patterns are possible which do not allow for easy caching.

4.10.4 Uniform Interface

The Uniform interface basically is what Hypermedia Multi-Agent Systems focus on; it nevertheless is very important.

4.10.5 Layered System

The Layered System constraint is easily fullfiled by using the HTTP protocol between artifact and agent. HTTP could also be used for communication among agents, however since HTTP can only be used for bidirectional communication initated by a client to a server, this would violated the clear separation between client and server.

4.10.6 Code-On-Demand

As simple reflex agents are based on rules, it would be easy for an artifact server to provide the agent with new rules on how to handle the specific resources provided by this server.

References

1 S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.
2 A. Ciortea, S. Mayer, F. Gandon, O. Boissier, A. Ricci, and A. Zimmermann, "A decade in hindsight: the missing bridge between multi-agent systems and the world wide web", in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2019.
3 A. Ciortea, O. Boissier, and A. Ricci, "Engineering world-wide multi-agent systems with hypermedia", in *International Workshop on Engineering Multi-Agent Systems*. Springer, 2018,pp. 285–301.
4 R. T. Fielding, "Architectural styles and the design of network-based software architectures". University of California, Irvine, 2000, vol. 7.

4.11 The Value of Knowledge: Opportunistic Knowledge Sharing

Valentina Tamma (University of Liverpool, GB) and María Poveda-Villalón (Technical University of Madrid, ES)

Agents are autonomous software entities that perform actions given the perception of their environment, and their prior knowledge. Agents exploit knowledge to different extents: from making sense of raw data from sensors, to exploiting implicit or explicit knowledge when cooperating with others. We focus on agents that utilise knowledge to exhibit autonomy and rationality and inhabit an ecosystem in which they seamlessly interact with other agents and humans.

Such ecosystem is necessarily knowledge driven: agents and the environment produce and consume knowledge, which is used to respond proactively and autonomously to the events occurring in the environment [7]. The agents' knowledge is expressed using different formalisms, is modelled in an agent's *ontology* to support communication.

Both agents and the environment are not static, but constantly evolve, with knowledge being produced or discarded, agents joining coalitions or operating independently in an *opportunistic* and *transient fashion.* The knowledge that agents produce and consume is an asset in its own right, has a measurable value quantified with respect to the tasks it supports. Therefore, agents need to treat knowledge as another resource they can made use of, and decide if, when and how to share it. However, sharing knowledge has both benefits and costs, and an agent acting rationally needs to evaluate these factors before deciding whether to share their knowledge [2]. In addition, an agent should be able to assess when it is in its interest to: 1) disclose all or part of its knowledge; 2) evolve this knowledge and align it with the knowledge of other agents in its environment; or 3) refuse to cooperate with other agents as it would be too costly in terms of disclosing its knowledge or evolving it.

The Semantic Web is, from this perspective, the perfect infrastructure for enabling this MAS knowledge-based ecosystem. Not only it provides the protocols that enable the sharing of fragments of knowledge (in the form of RDF triples or OWL axioms), but the wealth of existing vocabularies published in ontology repositories such as Linked Open Vocabularies [6], Bioportal [10] or in the LOD cloud promotes diversity in the representation of agent knowledge.

[10] https://bioportal.bioontology.org

Typically, theses vocabularies are used to annotate the Linked Data made available by SPARQL endpoints, or other publication formats, with the concepts and relations defining the domain at hand. In addition, some metadata is provided by complementary vocabularies as Provenance, Void, etc.

The idea of the Web infrastructure as a mechanism to share data using well defined standards and protocols has been extended to include physical objects, which are thus given a URI and can interact by using the HTTP protocol. This has motivated the specification of the network protocols underlying the Internet of Things (IoT) and the idea that Web applications consuming data from disparate devices. However, the IoT has never reached full harmonisation: the Web of Things (WoT) has emerged as an effort to counter fragmentation by proposing an application layer for the IoT based on web standards, such as standardised metadata and data transmission protocols (e.g.REST) to facilitate integration across IoT platforms and application domains.

Therefore, the Web is increasingly seen as an infrastructure for data production and consumption, where data is provided in manifold ways: directly as Linked Data, but also as a reference to the data source. For example in the WoT the data origin is described, and possibly transformed and exposed by SPARQL endpoints [1].

However, for this infrastructure to work efficiently, it requires the definition of fine tuned protocols or interaction methods to support decentralised collaborative approaches to knowledge sharing, where agents can agree on the terms to share [5, 3], or to evolve their knowledge to respond to errors [4].

In this context, the data might need to be translated into RDF in a posterior phase, following a set of mappings from the original sources to the target ontologies. In this context, it would be of interest of the WoT and semantic web communities to work towards the standardisation of mapping languages to be extended to WoT scenarios. In addition, the role of top-level ontologies should be analysed together with the use of complementary vocabularies (privacy models, access rights, provenance, behaviour, contracts, etc.)

In order to deliver such knowledge based ecosystem we need to address a number of challenges, some of which are aligned with those identified for this Dagstuhl seminar:

(i) Providing a vocabulary ecosystem configuration, that allows agents to identify and define 1) the minimum vocabulary or vocabularies needed, and 2) the possible relations amongst them to support discovery, access and interoperability amongst autonomous agents operating in the WoT;
(ii) Defining a more nuanced notion of "knowledge privacy", where agents make a value-based decision over the knowledge they share;
(iii) Establishing mechanisms for reaching semantic interoperability dynamically, that are opportunistic, and transaction based;
(iv) Representing and reasoning with uncertain and partial knowledge to model other agents.

References

1 A. Cimmino, M. Poveda-Villalón, and R. García-Castro . ewot: A semantic interoperability approach for heterogeneous iot ecosystems based on the web of things. *Sensors*, 20(3), 2020.
2 P. Doran, V. A. M. Tamma, T. R. Payne, and I. Palmisano. Dynamic selection of ontological alignments: A space reduction mechanism. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009*, pages 2028–2033, 2009.
3 J. Euzenat. Interaction-based ontology alignment repair with expansion and relaxation. In *IJCAI 2017-26th International Joint Conference on Artificial Intelligence*, pages 185–191. AAAI Press, 2017.

4 E. Jimenez Ruiz, T. R. Payne, A. Solimando, and V. A. Tamma. Limiting logical violations in ontology alignment through negotiation. 2016.
5 T. R. Payne and V. Tamma. A dialectical approach to selectively reusing ontological correspondences. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 397–412. Springer, 2014.
6 P.-Y. Vandenbussche, G. A. Atemezing, M. Poveda-Villalón, and B. Vatant. Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. *Semantic Web*, 8(3):437–452, 2017.
7 M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10:115–152, 1995.

4.12 Signifiers for Autonomous Agents: Perceiving and Acting in Affordance-Rich Hypermedia Environments

Danai Vachtsevanou (Universität St. Gallen, CH)

The Web's immense capacity of interaction possibilities is both what empowers users and what imposes on them the challenge of handling and making the most of such abundance of heterogeneous resources. Since the 1960s, affordance theory [1] has influenced many applied fields in modeling and designing possibilities of action offered to people by virtual and physical resources[11]. The Web environment itself is instilled with such affordances due to the mechanism that lands at the very depths of its heart: *hypermedia*. Hypermedia binds together information and controls such that the information becomes the affordance through which people and automated clients (or even autonomous agents) obtain choices and engage in interactions[12]. However, as the Web expands its uniform interface to accommodate users with varying abilities (i.e. different classes of autonomous agents) and resources whose capabilities are exposed through different mediums (i.e. through physical, virtual or mixed reality interfaces), we need to further investigate the following: a) what type of information is required for describing interactions with heterogeneous resources, and b) when and how this information should be rendered with respect to user abilities and objectives. Specifically, for diminishing the gap between the objectives and actions of software agents, we take inspiration from human-computer interaction towards the design of a new class of information carriers: a new class of *signifiers* [3] for autonomous agents on the Web.

The separation of concerns between affordances and signifiers reduces the coupling between the design of affordances and the design of perceptible information about affordances. This could be beneficial particularly for autonomous agents on the Web. Affordances express *which agent abilities and artifact*[13] *capabilities are compatible with each other* and *when*. Affordances can be expressed through *relationships* that are checked at run-time based on the temporal complementarity of an agent's abilities, an artifact's capabilities, and their state within their shared environment. On the other hand, signifiers convey to an agent

[11] We refer the interested reader to [2] for an overview of the fields that adopt affordance theory in design and applications.
[12] https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven
[13] Artifacts represent "resources and tools that agents can dynamically instantiate, share and use to support their individual and collective activities" [4].

clear and unambiguous cues on *how to exploit an emerging affordance and how relevant this interaction is for the agent* (e.g. based on the agent's intentions). By treating them as separate abstractions, affordances and signifiers can be modified and monitored independently by agents that have different interests and/or access rights (e.g. an agent may have the permission to update a signifier but not the related affordance).

Affordances and signifiers enable agents to pick up from their environment only the minimal information that is most relevant for interaction (i.e. *principle of economical perception*[1]), thus facilitating them to cope with large-scale environments. Specifically, agents' percepts could be adjusted quantitavely and qualitatively through limiting the set of perceived signifiers to one that maps to currently exploitable and prioritized affordances. Furthermore, since exploiting an affordance may lead to the perception of information about a new set of exploitable affordances, agents are given the chance to progressively explore their environment based on their intentions and take advantage of newly-discovered action opportunities. This step-wise navigation decouples further the agents from their environment, allowing both to evolve independently at run-time. To this end, it is interesting to investigate the following:

- How to model and represent affordances as relationships between autonomous agents and their hypermedia environment.
- How to model and represent signifiers for autonomous agents in hypermedia environments.
- How to enable autonomous agents, artifact designers and environment designers to publish, share and modify signifiers.
- How to enable autonomous agents to perceive signifiers based on the principle of economical perception.
- How to design mechanisms for dynamically adjusting the salience of signifiers such as to properly invite autonomous agents to interact.

References

1. J. J. Gibson. *The ecological approach to visual perception: classic edition.* Psychology Press, 1979.
2. L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater and J. Santos-Victor. *Affordances in psychology, neuroscience, and robotics: A survey.* IEEE Transactions on Cognitive and Developmental Systems, 10(1), 4-25. 2016.
3. D.A. Norman. *The Design of Everyday Things.* BasicBooks, 2013.
4. A. Ricci, M. Piunti and M. Viroli. *Environment programming in multi-agent systems: an artifact-based perspective.* Autonomous Agents and Multi-Agent Systems, 23(2), 158-19. 2011.

5 Overview of Demonstrators

5.1 A Toolchain for Enabling Process Mining from IoT Data

Ronny Seiger (Universität St. Gallen, CH) Andrea Burattin (Technical University of Denmark, DK) and Barbara Weber (Universität St. Gallen, CH)

Process Mining (PM) as a subdiscipline of Business Process Management (BPM) focuses on the discovery, anaylsis and improvement of business processes based on the digital traces (event logs) of the corresponding process executions monitored and managed by a BPM system [1]. While these systems usually exist within enterprises to manage various kinds of digital commercial and organizational processes, the availability of a BPM system in Internet of Things (IoT) environments or cyber-physical systems (CPS) cannot always be assumed. Thus, the large body of process mining methods and techniques developed by the PM community is not applicable "as-is" to this kind of high-level processes–although these processes clearly exist in CPS/IoT and would benefit from a PM-based analysis [2]. On the other hand, data produced by the sensors, devices and machines of an IoT environment is usually not process-aware and too fine-grained (low-level) to be a suitable basis for process mining.

This demo presents a toolchain for enabling process mining from IoT data of a smart factory [3]. Using recorded and replayed event streams of raw IoT data from a smart factory, we show how to use stream processing/complex event processing (CEP) as core technology to aggregate, abstract and correlate these low-level event streams to higher level (business) process events [4]. These process events are transformed into the XES standard to enable 1) offline process mining from a persisted event log, and 2) online process mining on process event streams [5]. With standard process mining tools and techniques, the underlying processes and statistics can then be discovered, conformance regarding a normative process model can be checked, and processes can be enhanced. These techniques may also be suitable for visualizing and analyzing interactions and collaboration (e.g., as social networks) among multiple agents in autonomous systems–not necessarily relying on the presence if a BPM system to manage the process executions.

References

1 Wil Van Der Aalst. *Data science in action*. Process mining, Springer, 2016.
2 Christian Janiesch, Agnes Koschmider, Massimo Mecella, Barbara Weber, Andrea Burattin, Claudio Di Ciccio, Giancarlo Fortino, Avigdor Gal, Udo Kannengiesser, Francesco Leotta and others. *The internet of things meets business process management: A manifesto*. IEEE Systems, Man, and Cybernetics Magazine, 6(4) :34-44, 2020.
3 Lukas Malburg, Ronny Seiger, Ralph Bergmann and Barbara Weber. *Using physical factory simulation models for business process management research*. International Conference on Business Process Management, Springer, Cham, 2020
4 Ronny Seiger, Francesca Zerbato, Andrea Burattin, Luciano García-Bañuelos and Barbara Weber. *Towards iot-driven process event log generation for conformance checking in smart factories*. 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW), 2020.
5 Andrea Burattin, Martin Eigenmann, Ronny Seiger and Barbara Weber. *MQTT-XES: Real-time telemetry for process event data*. BPM 2020 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track, 2020.

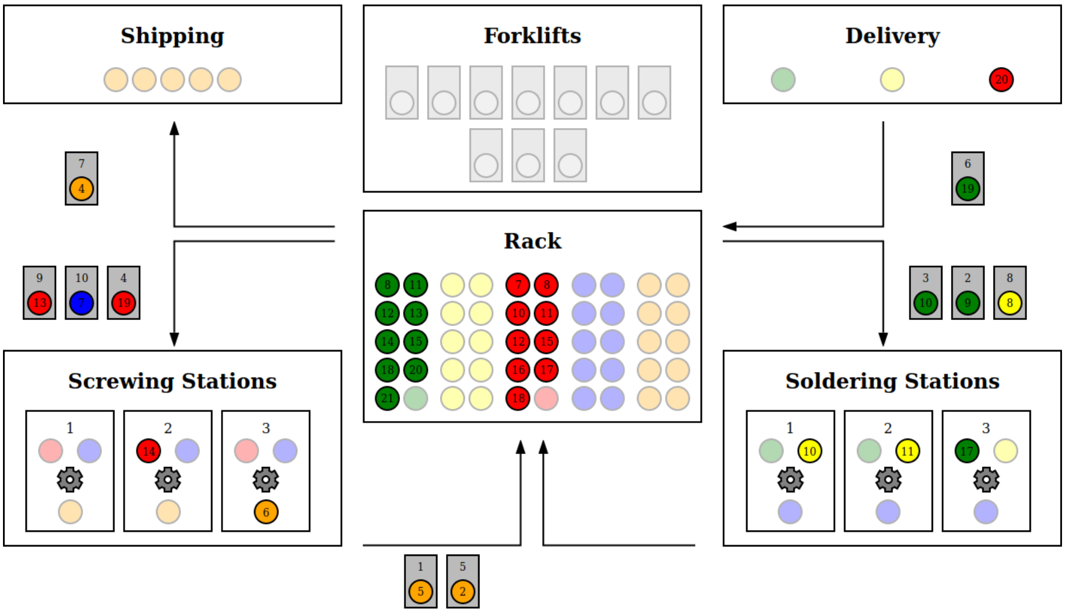


Figure 3 The base parts appear at the delivery station. They are transported to the according workstations and stored in the Rack inbetween. In the end the IoT boards get shipped.

5.2 Linked Data-Fu – A Manufacturing Demo

Daniel Schraudner (Universität Erlangen-Nürnberg, DE) and Andreas Harth (Fraunhofer IIS – Nürnberg, DE)

Main reference Daniel Schraudner, Victor Charpenay: An HTTP/RDF-Based Agent Infrastructure for Manufacturing Using Stigmergy. ESWC (Satellite Events) 2020: 197-202
URL https://doi.org/10.1007/978-3-030-62327-2_34

Linked Data-Fu is an end-to-end data processing system for data integration and system interoperation with Linked Data. Linked Programs are Notation3-based specifications for accessing, processing and changing Linked Data, based on logical rules and production rules. Linked Programs can be evaluated with ldfu.[1]

We implement simple reflex agents by executing ldfu in a cylce. The agents can explore and perceive their environment by sending HTTP GET requests where the Notation3 rules control which GET requests are sent out and thus control the exploring behavior of the agents. As the Notation3 rule correspond to the agents' reflex rules, they also guide the agents' actions, i. e. whether and how they act upon their environment by sending unsafe HTTP requests.

The simple reflex agents are used for controlling automated guided vehicles and workstations in a simulated manfacturing scenario coordinating themselves using stigmergy. The task in the scenario is to manufacture IoT boards that are build out of two parts, one of those two parts being again build out of two other parts [2]. An overview of the process can be seen in Figure 1.

References

1 Harth A. and Käfer T. "Specifying and Executing Application Behaviour with Condition-Request Rules". ISWC2017 workshop on Decentralizing the Semantic Web, October 22, 2017, Vienna, Austria.

2 Schraudner, D. and Charpenay, V. "An HTTP/RDF-Based Agent Infrastructure for Manufacturing Using Stigmergy". In: European Semantic Web Conference. Springer, Cham, 2020. S. 197-202.

5.3 Stigmergy for Simple Reflex Agents in Self-Organizing Transportation

Sebastian Schmid (Universität Erlangen-Nürnberg, DE)

In our demo, we show an approach to use simple reflex agents [1] with stigmergy to fulfill a basic transportation task in a self-organizing manner. Stigmergy is defined as indirect communication between agents through changing and evaluating their common environment [4]. It is inspired by social insects, e.g. ants and their creation of trails with the help of placed pheromones [3]. Our approach is motivated by the pickup-and-delivery-problem [2] and shall provide new thoughts on how to handle dynamic and stochastic influences. We focus on decentralized agent interaction, a simple rule set, restriction of their perception to local surroundings and on the environment's emphasized role as sole possibility for indirect communication between agents.

In our simulation, randomly moving transporters wander around a defined shop floor with a fixed number of stations. Here, the transporters are simple reflex agents using stigmergy. Their task is to pick up and deliver randomly generated, colored items to a corresponding station, while following their simple set of rules and the restrictions from above. Agents share their knowledge about destinations of transportation requests of colored items by creating uniform marks in their common environment. When they come across a suitable station during their random walk and can deliver their carried item successfully, the transporters place a corresponding mark in their environment around the station. When they carry an item and perceive an already placed, corresponding color mark in their own surroundings, they follow these marks and restrict their search area to the marked zone instead of randomly wandering around. Hence, the transporters cooperate indirectly to show each others paths to find the stations they are looking for. They perceive, share and amend each others knowledge via these stigmergy markers in their environment.

We compare these simple reflex agents in two scenarios, one with the usage of stigmergy and one without, that have been implemented in GAMA, a modeling and simulation environment, with respect to the amount of delivered items over time and the mean time to deliver an item. We observe that the agents create a new emergence in the given setup in terms of performance improvement when they use stigmergy.

This work was funded by the German Federal Ministry of Education and Research through the MOSAIK project (grant no. 01IS18070A).

References

1 Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 2009
2 Martin Savelsbergh and Marc W. P. Sol. *The General Pickup and Delivery Problem*. Transportation Science, 29.1: 17-29, 1995
3 Marco Dorigo, Mauro Birattari and Thomas Stützle. *Ant Colony Optimization*. IEEE Computational Intelligence Magazine, 1.4: 28-39, 2006
4 Francis Heylighen. *Stigmergy as a universal coordination mechanism I: Definition and components*. Cognitive Systems Research, 38: 4-13, 2016

5.4 Exploring Agent-Oriented Programming Abstractions in JavaScript with JS-son

Timotheus Kampik (University of Umeå, SE)

Joint work of Timotheus Kampik, Andrés Gomez, Andrei Ciortea, Simon Mayer
URL https://github.com/TimKam/JS-son

According to a recent community report [4], more work is needed to bridge the gap between academic agent-oriented programming and mainstream programming approaches that are employed by the industry. To work towards this objective, we are developing the JS-son JavaScript library [2] (prototype) for implementing cognitive agents. JS-son supports various reasoning loops that resemble the classical belief-desire-intention approach [1] to different extents. It is possible to deploy JS-son agents to a broad range of environments, for example servers, clients, and data science environments (Jupyter notebooks[14]). Our latest work[15] integrates JS-son agents with constrained devices and the W3C Web of Things Scripting API [3]. However, an open question remains: how can we design agent-oriented abstractions so that they are as pragmatic and developer-friendly as possible? To answer this question, we suggest that it is necessary to take a step back, and explore agent-oriented programming concepts one-by-one, from the perspective of an industry software engineer.

References

1 Bratman, M. *Intention, Plans, and Practical Reason.* Cambridge, MA: Harvard University Press, 1987.
2 Kampik, T., and Nieves, J. C. Js-son – a lean, extensible javascript agent programming library. In *Engineering Multi-Agent Systems* (Cham, 2020), L. A. Dennis, R. H. Bordini, and Y. Lespérance, Eds., Springer International Publishing, pp. 215–234.
3 Kis, Z., Peinter, D., Aguzzi, C., Hund, J., and Nimura, K. Web of Things (WoT) Scripting API, W3C Working Group Note 24 November 2020. W3C Working Group Note, World Wide Web Consortium (W3C), Nov. 24 2020.
4 Mascardi, V., Weyns, D., Ricci, A., Earle, C. B., Casals, A., Challenger, M., Chopra, A., Ciortea, A., Dennis, L. A., Díaz, A. F., El Fallah-Seghrouchni, A., Ferrando, A., Fredlund, L.-r., Giunchiglia, E., Guessoum, Z., Günay, A., Hindriks, K., Iglesias, C. A., Logan, B., Kampik, T., Kardas, G., Koeman, V. J., Larsen, J. B., Mayer, S., Méndez, T., Nieves, J. C., Seidita, V., Teze, B. T., Varga, L. Z., and Winikoff, M. Engineering multi-agent systems: State of affairs and the road ahead. *SIGSOFT Softw. Eng. Notes 44*, 1 (Mar. 2019), 18–28.

[14] `https://jupyter.org/`
[15] To be demonstrated at AAMAS '21, see: `https://youtu.be/MUhUuqd2jt0`.

5.5 Interactive and Collaborative Programming of Interoperable Agents Using Jacamo-Web, Jacamo-rest and Camel-Jacamo components

Cleber Jorge Amaral (Universidade Federal de Santa Catarina, BR), Jomi Fred Hübner (Federal University of Santa Catarina, BR), and Timotheus Kampik (University of Umeå, SE)

Joint work of Otavio A. Matoso; Luis P. A. Lampert, Mateus Conceição, Sergio P. Bernardes, Maicon R. Zatelli, Marcelo L. de Lima, Stephen S. Cranefield
Main reference Cleber Jorge Amaral, Timotheus Kampik, Stephen Cranefield: "A Framework for Collaborative and Interactive Agent-oriented Developer Operations", in Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020, pp. 2092–2094, International Foundation for Autonomous Agents and Multiagent Systems, 2020.
URL https://dl.acm.org/doi/abs/10.5555/3398761.3399086
Main reference Otávio Arruda Matoso, Luis Pedro Arenhart Lampert, Jomi Fred Hübner, Mateus Conceição, Sérgio Pereira Bernardes, Cleber Jorge Amaral, Maicon Rafael Zatelli, Marcelo L. de Lima: "Agent Programming for Industrial Applications: Some Advantages and Drawbacks", CoRR, Vol. abs/2006.05613, 2020.
URL https://arxiv.org/abs/2006.05613
Main reference Cleber Jorge Amaral, Jomi Fred Hübner, Timotheus Kampik: "Towards Jacamo-rest: A Resource-Oriented Abstraction for Managing Multi-Agent Systems", CoRR, Vol. abs/2006.05619, 2020.
URL https://arxiv.org/abs/2006.05619

To facilitate the vision of autonomous agents that interact on the Web, we present web-based and resource-oriented interfaces to the JaCaMo framework for developing Multi-Agent Systems (MAS). These abstractions are implemented by jacamo-rest, a JaCaMo extension that provides REST endpoints to a server-based JaCaMo instance, as well as Apache Camel components that allow agents to act as clients on the Web. Jacamo-rest's endpoints allow external entities to interact with a JaCaMo MAS' internal agents and artifacts, and also support the administration of a MAS by external applications, for instance, to create, modify, and send commands to agents. The Camel-jacamo component, in turn, enables agents of the MAS to connect to external entities (both agents and artifacts), providing proper abstractions to them and leaving the complexity of the integration to the Apache Camel middleware. Using jacamo-rest and camel-jacamo, a MAS can be consumed by external entities on the Web and the MAS's entities can consume services on the Web. We also introduce jacamo-web, an interactive and collaborative development environment that illustrates how to consume jacamo-rest endpoints, and an industrial application that shows how to exploit camel-jacamo components for connecting agents with heterogeneous systems.

5.6 Hypermedia Multi-Agent Systems

Andrei Ciortea (Universität St. Gallen, CH), Olivier Boissier (Ecole des Mines – St. Etienne, FR), Fabien Gandon (INRIA – Sophia Antipolis, FR), Simon Mayer (Universität St. Gallen, CH), Alessandro Ricci (Università di Bologna, IT), and Antoine Zimmermann (Ecole des Mines – St. Etienne, FR)

Main reference Andrei Ciortea, Olivier Boissier, Alessandro Ricci: "Engineering World-Wide Multi-Agent Systems with Hypermedia", in Proc. of the Engineering Multi-Agent Systems – 6th International Workshop, EMAS 2018, Stockholm, Sweden, July 14-15, 2018, Revised Selected Papers, Lecture Notes in Computer Science, Vol. 11375, pp. 285–301, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-030-25693-7_15

Hypermedia-driven APIs are promoting the development of a new generation of dynamic, open, and long-lived systems on the Web that would profit from agent-based solutions, but there is still a conceptual gap between architectures for MAS and the Web Architecture [4]. Our objective is to bridge this gap through a design rationale for engineering Web-based MAS that are aligned with the Web architecture and can inherit the non-functional properties of the Web as an Internet-scale, open, and long-lived system. Our approach draws from the design rationale behind the Web architecture, which is captured by the REST architectural style [5], to define a hypermedia-based uniform interface for MAS. In our approach, the agent environment is a first-class abstraction [6] and uses hypermedia to enable uniform interaction among heterogeneous entities across any conceptual dimensions in MAS (e.g., agent, environment, organization). The hypermedia-based uniform interface reduces coupling and enhances the scalability, openness, and evolvability of MAS. We refer to this class of Web-based MAS as *Hypermedia MAS*.

During the seminar, we presented a demonstrator for Hypermedia MAS based on [3]. The demonstrator shows *Belief-Desire-Intention (BDI)* agents [2] in a distributed, open, and evolvable hypermedia-based agent environment. Given a set of semantic models and a single entry URI into the system, the agents are able to achieve their design objectives by navigating the hypermedia to discover, create, perceive, and act on artifacts. The demonstrator was built using Yggdrasil[16], the JaCaMo platform [1], and the hardware infrastructure provided by the *Chair for Communication- and Interaction-based Systems* at the University of St.Gallen[17].

References

1 Boissier O., Bordini R.H., Hübner J.F., Ricci A., and Santi A. 2013. Multi-agent oriented programming with JaCaMo, Science of Computer Programming, Volume 78, Issue 6, Pages 747-761, ISSN 0167-6423, `https://doi.org/10.1016/j.scico.2011.10.004`.
2 Bordini R.H., Hübner J.F., and Wooldridge M. 2007. Programming multi-agent systems in AgentSpeak using Jason. John Wiley & Sons.
3 Ciortea A., Boissier O., and Ricci A. 2019. Engineering World-Wide Multi-Agent Systems with Hypermedia. In: Weyns D., Mascardi V., Ricci A. (eds) Engineering Multi-Agent Systems. EMAS 2018. Lecture Notes in Computer Science, vol 11375. Springer, Cham. `https://doi.org/10.1007/978-3-030-25693-7_15`
4 Ciortea A., Mayer S., Gandon F., Boissier O., Ricci A., and Zimmermann A. 2019. A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World

[16] Yggdrasil is a platform for Hypermedia MAS based on the *Agents & Artifacts* meta-model [6]: `https://github.com/interactions-hsg/yggdrasil`
[17] `https://interactions.ics.unisg.ch/`

Wide Web. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1659–1663.
5 Roy T. Fielding and Richard N. Taylor. 2002. Principled design of the modern Web architecture. ACM Transactions on Internet Technology 2, 2 (May 2002), 115–150. DOI: https://doi.org/10.1145/514183.514185
6 Ricci, A., Piunti, M., and Viroli, M. 2011. Environment programming in multi-agent systems: an artifact-based perspective. Autonomous Agents and Multi-Agent Systems 23, 158–192. https://doi.org/10.1007/s10458-010-9140-7

5.7 Artifact Manuals for the Continuous Acquisition of Agent Behaviors in Hypermedia Environments

Danai Vachtsevanou (Universität St. Gallen, CH)

Main reference Danai Vachtsevanou, Philip Junker, Andrei Ciortea, Iori Mizutani, Simon Mayer: "Long-Lived Agents on the Web: Continuous Acquisition of Behaviors in Hypermedia Environments", in Proc. of the Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pp. 185–189, ACM / IW3C2, 2020.
URL http://dx.doi.org/10.1145/3366424.3383537

In Hypermedia Multi-Agent Systems [1], human and software agents are situated in an environment designed as a distributed hypermedia application. All autonomous agents can proactively navigate the hypermedia environment to discover and use Web resources modeled as artifacts [2]. Based on work of the W3C Web of Things initiative[18] and the new Thing Description Recommendation[19], autonomous agents can leverage the hypermedia environment to dynamically discover and map individual interaction affordances of Web Things[20] to artifact operations (e.g. by using browser artifacts [1]), while staying decoupled from specific APIs. However, operations remain atomic targeting low-level interactions, while the task of addressing more complex interactions is always left to agents.

In this demonstrator, we showcase how autonomous agents can dynamically extend their repertoire of complex interactions. By navigating the hypermedia environment, Jason[21] agents can discover artifact manuals [3] that provide explicit descriptions of operating instructions in the form of usage protocols. Usage protocols are mapped to plans in AgentSpeak [4] which agents can directly use to exploit Web resources based on newly acquired high-level behaviors. As a result, agents can engage in complex interactions in a way that balances between inflexible hardcoded behaviors and time-consuming automated planning: It becomes possible to search for usage protocols (e.g. by means of a hypermedia search engine [5]), consult manual repositories or even share directly protocols with other agents. Ultimately, bringing usage protocols to the hypermedia environment enables agents to adjust their abilities to new goals and available resources, thereby promoting agents' longevity in open and flexible systems.

[18] https://www.w3.org/WoT/wg/
[19] https://www.w3.org/TR/wot-thing-description/
[20] https://www.w3.org/TR/wot-architecture#sec-web-thing
[21] http://jason.sourceforge.net/

References

1 Ciortea A., Boissier O., and Ricci A. 2019. *Engineering World-Wide Multi-Agent Systems with Hypermedia.* In: Weyns D., Mascardi V., Ricci A. (eds) *Engineering Multi-Agent Systems.* EMAS 2018. Lecture Notes in Computer Science, vol 11375. Springer, Cham. https://doi.org/10.1007/978-3-030-25693-7_15

2 Ricci, A., Piunti, M., and Viroli, M. 2011. *Environment programming in multi-agent systems: an artifact-based perspective.* Autonomous Agents and Multi-Agent Systems 23, 158–192. https://doi.org/10.1007/s10458-010-9140-7

3 Ricci, A., Piunti, M., and Viroli, M. 2009. *Externalisation and internalization: A new perspective on agent modularisation in multi-agent system programming.* In International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems (pp. 35-54). Springer, Berlin, Heidelberg.

4 Bordini, R. H., Hübner, J. F., and Wooldridge, M. 2007. *Programming multi-agent systems in AgentSpeak using Jason (Vol. 8).*John Wiley & Sons.

5 Bienz, S., Ciortea, A., Mayer, S., Gandon, F., and Corby, O. 2019. *Escaping the streetlight effect: Semantic hypermedia search enhances autonomous behavior in the web of things.* In Proceedings of the 9th International Conference on the Internet of Things (pp. 1-8).

5.8 Query Execution over Linked Data Fragments

Ruben Taelman (Ghent University, BE)

A large amount of Linked Data is available on the Web, which can be published and queried in different ways. In this demonstration, we explained how Linked Data Fragments offers different ways to publish RDF datasets on the Web via hypermedia controls. Different Linked Data Fragments offer different trade-offs between server and client effort when querying over them. Triple Pattern Fragments [1] is one type of Linked Data Fragments that exposes a triple pattern querying interfaces, which leads to a low publication cost, but leads to a larger querying effort for clients. We showed how a client-side SPARQL query engine (Comunica [2]) can interpret such controls in order to intelligently delegate query effort between server and client, and additionally allows federated querying across multiple heterogeneous Linked Data Fragments. The Linked Data Fragments framework therefore offers a path to query execution over the Web of data on Web-scale.

References

1 Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, and Ben De Meester, Gerald Haesendonck and Pieter Colpaert. *Triple Pattern Fragments: a low-cost knowledge graph interface for the Web.* Journal of Web Semantics, vol. 37, pp. 184–206, 2016

2 RubenTaelman, Joachim Van Herwegen, Miel Vander Sande and Ruben Verborgh. *Comunica: a modular SPARQL query engine for the web.* International Semantic Web Conference, pp. 239–255, Springer, 2018

5.9 WoTDL2API: Webifying Heterogenous IoT Devices

Mahda Noura (Technische Universität Chemnitz)

Joint work of Mahda Noura, Sebastian Heil, Martin Gaedke
Main reference Mahda Noura, Sebastian Heil, Martin Gaedke: "Webifying Heterogenous Internet of Things Devices", in Proc. of the Web Engineering – 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11-14, 2019, Proceedings, Lecture Notes in Computer Science, Vol. 11496, pp. 509–513, Springer, 2019.
URL http://dx.doi.org/10.1007/978-3-030-19274-7_36

The advancements in the Internet of Things (IoT) domain have increased interest in application development with heterogenous smart devices. However, the devices on the market are not interoperable and support different standards and communication protocols (Zigbee, RFID, Bluetooth, or even custom protocols). Enabling application development employing different protocols require interoperability between the different types of heterogenous devices that coexist in the IoT ecosystem. Therefore, it is tedious, time-consuming and error-prone to develop applications for a particular use case with heterogenous devices.

In this demo, we showcase WoTDL2API [2] using IoT devices from the smart home domain. WoTDL2API automatically generates a running RESTful API based on the popular OpenAPI specification and integrating with the existing OpenAPI code generation toolchain. The devices are described according to the Web of Things Description Language (WoTDL) [1]. This solution provides interoperability between the devices by wrapping IoT devices with a Web-based interface enabling easier integration with other platforms.

References

1 Noura, Mahda and Gaedke, Martin. Wotdl: Web of things description language for automatic composition. 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pages 413–417, 2019.

2 Noura, Mahda and Heil, Sebastian and Gaedke, Martin. Webifying heterogenous internet of things devices. International Conference on Web Engineering, pages 509–513, 2019.

5.10 BOLD: A Platform for Evaluating Linked Data Agents

Victor Charpenay (Mines Saint-Étienne, FR)

URL https://github.com/bold-benchmark/bold-server

The BOLD demo presented at Dagstuhl introduces a Web server designed for quickly prototyping Linked Data environments. This server was originally used to provide a Linked Data interface to simulated buildings, hence the name BOLD (Building on Linked Data). It is however a configurable software component that could be used on a variety of scenarios. The BOLD server can be found online[22].

The BOLD server has two main components. Its first component is a front-end to an RDF dataset, receiving agents' actions and updating the RDF dataset accordingly. The HTTP interface to the dataset is in fact compliant with the SPARQL Graph Store

[22] `https://github.com/bold-benchmark/bold-server`

protocol, standardized by the W3C[23]. The BOLD server's second component is a SPARQL rule production system, to simulate e.g. occupancy and sunlight in a building or other spontaneous environmental changes. Once a simulation is started, rules are re-evaluated at a fixed rate (100ms by default). The BOLD server allocates a thread pool to answer concurrent agent requests and a separate thread to evaluate rules in parallel.

During the demo, the environment run by the BOLD server was a simulated building. Agents must continuously monitor building data and fix "faults". A fault is a light that is on in a room that is not occupied or whose average illuminance level is above a threshold set by the occupant. Agents can navigate between rooms through hypermedia, in order to discover devices such as light switches, occupancy sensors and illuminance sensors. In this scenario, fixing faults requires only few updates (`PUT` requests). A naive agent retrieving all sensor values at a high rate would however have to perform many `GET` requests, while more sophisticated agents could learn trends and use inference to reduce the number of requests.

A baseline agent has been implemented using Linked-Data-Fu (ldfu)[24]. Ldfu is a utility designed for reading and writing Linked Data at a Web scale (a ldfu agent may manage billions of RDF quads in its internal state). The Linked Data programs executed by the ldfu agent for the building scenario are available online[25] To illustrate the integration with existing agent platforms through artifacts, an equivalent agent has been implemented with JaCaMo[26]. The JaCaMo code base for BOLD is again available online[27].

6 Working Group Reports

6.1 Designing Agents and Multi-Agent Systems for the Web (Conceptual Mappings)

Cleber Jorge Amaral (Universidade Federal de Santa Catarina, BR), Andrei Ciortea (Universität St. Gallen, CH), Jérôme Euzenat (INRIA – Grenoble, FR), Fabien Gandon (INRIA – Sophia Antipolis, FR), Andreas Harth (Fraunhofer IIS – Nürnberg, DE), Jomi Fred Hübner (Federal University of Santa Catarina, BR), Tobias Käfer (KIT – Karlsruher Institut für Technologie, DE), Matthias Kovatsch (Huawei Technologies – München, DE), Mahda Noura (TU Chemnitz, DE), Cesare Pautasso (University of Lugano, CH), María Poveda-Villalón (Technical University of Madrid, ES), Alessandro Ricci (Università di Bologna, IT), and Valentina Tamma (University of Liverpool, GB)

This working group focused on aligning the terminology and determining conceptual mappings across the three areas targeted by this seminar: *Web Architecture and the Web of Things*, *Semantic Web and Linked Data*, and *Autonomous Agents and Multi-Agent Systems*. The discussions covered a broad range of topics, which we report briefly in what follows. One of

[23] https://www.w3.org/TR/sparql11-http-rdf-update/
[24] https://linked-data-fu.github.io/
[25] https://github.com/bold-benchmark/bold-agents
[26] https://github.com/jacamo-lang/jacamo
[27] https://github.com/bold-benchmark/bold-jacamo/

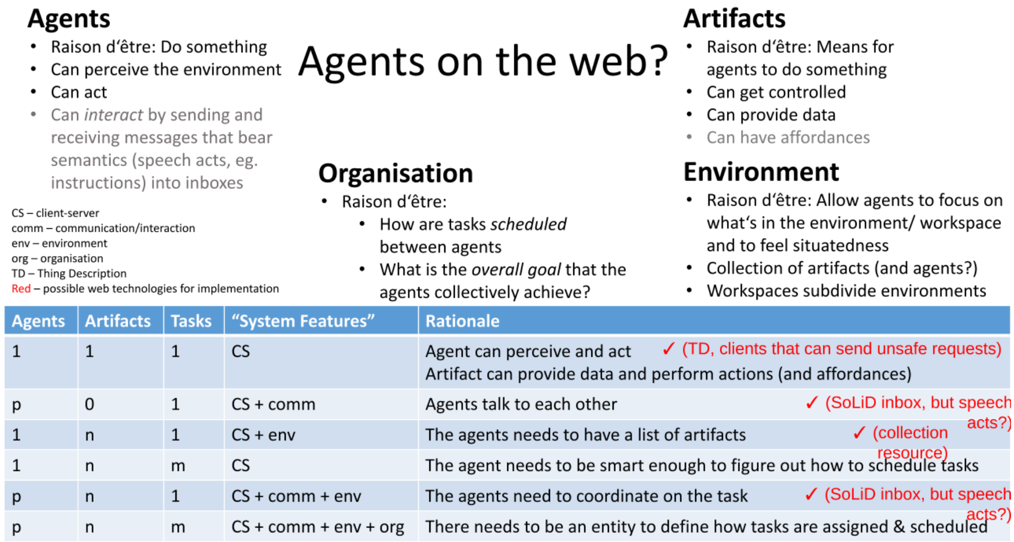


Figure 4 Slide used in the 3rd day as a concrete illustration.

the main outcomes from this working group was a general consensus to create a shared "live" demonstrator space that would allow to try out various ideas and technologies for Web-based multi-agent systems (MAS). The idea was well received by the seminar participants and resulted in an initiative to organize a series of follow-up demo-oriented meetings – where the shared demonstrator space could serve as a testbed for trying out ideas from other working groups as well. The first follow-up meeting was scheduled for July 2021.

6.1.1 Discussion Overview

The first day started out with collecting and discussing terms from the three targeted communities and focused on MAS-related terms. The discussion covered a broad range of topics across all four conceptual dimensions typically considered when engineering multi-agent systems (i.e., the *agent*, *environment*, *interaction*, and *organization* dimensions). Some of the emerging questions included: How to identify and represent agents on the Web? How to map agent communication and interaction mechanisms to HTTP-based communication? How to define and represent situatedness in Web environments? How to define norms and how to regulate autonomous behavior on the Web? While the answers to these questions were more preliminary given the short time available for discussion, the working group did not identify any fundamental problems that would prevent deploying any such mechanisms on the Web.

The second day continued the terminology discussion with a focus on Semantic Web terms. One conclusion was the necessity to define a common terminology for the three communities that would resolve colliding/misleading terms (e.g., "autonomous agent" vs. "user agent"). Various approaches were discussed for how MAS could be integrated with the Web, such as using the Web as an infrastructure for distributing MAS runtimes (i.e., MAS scale across the Web, but agents are not aware of the Web) or using the Web as an environment that agents can perceive and act upon. Other discussion topics included knowledge representation in the Semantic Web, defining boundaries for locality/situatedness on the Web, or what would be the minimal constraints for bringing agents to/on the Web. One conclusion was that more hands-on experience would help to bring more insight into these topics.

The third day started with a concrete illustration of how the various concepts and technologies discussed in the previous days could fit together (see Figure 4). As a starting point, we discussed high-level working definitions for the components of multi-agent systems: agents, artifacts, organisation, and environment. The next step was to motivate each component bottom-up. We started out with one agent that has to fulfil one task. Then, we defined different scenarios by varying the numbers of agents, artifacts and tasks. From the scenarios, we derived required system features: client-server interaction between agents and artifacts, inter-agent communication, artifact catalogue, task assignment and scheduling means. We note that not all scenarios require all features. Next, we checked for each scenario which Web technologies we could use today to build a (multi-)agent system for the scenario. This allowed us to identify in which scenarios we could apply agents based on Web technologies already available today and which scenarios represent gaps for future research and development. This discussion set the stage for proposing a shared "live" demonstrator space, which we present next.

6.1.2 A Call to Action

The working group made good progress on the ambitious task of aligning some of the core concepts and terms across the three targeted areas. The general consensus, however, was that a more hands-on approach is needed: in addition to identifying conflicting terminology and discussing concept mappings, it would help to build working prototypes and try things out. The demonstrators presented during the seminar's *Demos & Tech Sessions* (see Section 5) had already uncovered a diverse (and non-exhaustive) set of tools that could be used to experiment.

With the above motivation in mind, the working group ideated about setting up a shared "live" demonstrator space: a deployed, open, and geographically distributed hypermedia environment that could provide a testbed for trying out new ideas and identifying challenges (e.g., coping with different media types, coping with different vocabularies and ontologies in RDF-based environments). Several seminar participants have expressed interest in contributing to the shared demonstrator space and showed support for a series of follow-up demo-oriented meetings. At the moment of writing this report, the first meeting was scheduled for July 2021.

6.2 Affordances and Signifiers

Victor Charpenay (Mines Saint-étienne, FR), Mike Amundsen (MuleSoft LLC – San Francisco, US), Simon Mayer (Universität St. Gallen, CH), Julian Padget (University of Bath, GB), Daniel Schraudner (Universität Erlangen-Nürnberg, DE), and Danai Vachtsevanou (Universität St. Gallen, CH)

6.2.1 Overview

Hypermedia systems are distributed by nature. Each component of a hypermedia system should operate with a certain level of autonomy with respect to other components. This property makes autonomous agents an appropriate programming abstraction for developing hypermedia system components. Agents are indeed meant to make local decisions, without being able to perceive the entirety of their environment. At a Web scale, it is however likely that agents have to deal with a high volume/variety of information. When designing hypermedia environments for autonomous agents, an important question is: what should agents perceive in their hypermedia environment?

In this chapter, we address the question by adopting the governing principle of *economical perception*, which means that agents select from their environment only the minimal information that is most relevant for interaction. Two major concepts follow from this principle, advanced by J. J. Gibson [9]: affordances and signifiers. The first (affordances) are what the environment allows a particular agent to do or, in other words, what the environment *affords* to the agent, and the second (signifiers) are commonly perceivable features of the environment that convey affordances.

We introduce affordances and signifiers in more detail in the remainder of the chapter. We start in Sec. 6.2.2 with background on the original conceptualization of affordances introduced by J.J Gibson and later developed by Don Norman with signifiers. We also provide a discussion of social aspects of affordances. Then, Sec. 6.2.3 introduces more specific kinds of affordances on the Web, through hypermedia controls such as *links* and *forms*. The section also shows how hypermedia controls can allow Web agents to interact with physical objects on the Web of Things (WoT). Finally, Sec. 6.2.4 moves on to signifiers and their resolution (from a perceived entity to a list of interaction choices) in a way that is compliant with the principle of economic perception. Stigmergy, for instance, a form of interaction in which an agent's perception is limited to traces left by other agents, can be seen as a way to produce signifiers (the traces) associated with a fast signifier resolution mechanism, from perception to action.

6.2.2 Origin of Affordances and Signifiers

6.2.2.1 Affordances in Psychology and Design

The theory of affordances aims to describe how animals perceive and exploit possibilities for action in their environment. Introduced by the ecological psychologist J. J. Gibson, the concept of *affordance* expresses any action possibility offered by the environment that captures the animal-environment complementarity required for interaction [9]. More specifically, an affordance is commonly defined as *a relation between an animal and its environment that have consequences for behavior* [18, 20, 21]. A human agent can perceive and exploit the *toggleable* affordance of a lamp only if the agent's motor skills are compatible with the structural characteristics of the switch. The affordance, as the possibility of the agent to toggle the switch, does not emerge if these properties do not exhibit complementary values (e.g. in the case of an infant with insufficient strength). The agent-environment complementarity may also be dynamic if it depends on the states of the agent and the environment, e.g. if the hands of the agent are currently occupied.

Possibilities for action are in themselves objective as their existence does not presuppose their perception and interpretation but what agents perceive is subjective, as they have a single perspective on their environment. Affordances as relations between agents and entities thus gives rise to the principle of *economical perception* in interaction, which is presented as an enabler in coping with large-scale and affordance-rich environments. By providing all the necessery information for interaction without the need for time-consuming reasoning, affordances guide and enhance the behavioral economy of animals. Although the existence of an affordance enables a certain behavior, it does not necessarily imply the agent's engagement to such behavior [23].

Since its conception, affordance theory has influenced many applied fields such as autonomous robotics, computer vision and human-computer interaction. Norman, in particular, was interested in perceived affordances [28] in the context of Human-Computer Interaction (HCI) and the principles that outline human interaction for designers of physical and virtual environments. He focused on designing "everyday things" for which the user can intuitively perceive and exploit what is afforded to them [24].

Gradually, the interest of designers was shifted towards the concept of *signifier*[29] – a term adopted from the field of semiotics to denote *any perceivable cue (deliberate or accidental) that can be interpreted meaningfully to reveal information about affordances*. Although affordances determine which actions are possible for a user, signifiers are perceivable cues that convey appropriate behavior. As a result, both designers and users aim at increasing the number and salience of useful signifiers in the environment, and also, in optimizing the position of signifiers within the environment based on their needs and preferences [24].

6.2.2.2 Social Affordances

The conventional view on affordances pertains to physical objects, but the concept can also be applied to the intangible, in the form of norms, to view them as social affordances, in which first-order norms inform agent action and higher-order norms inform how to bring about norm change [5]. The norm literature draws a distinction between implicit norms, which are internal to an agent, and whose nature may possibly only be inferred via observation of agent (inter)action and explicit or referenceable norms, that have a representation, typically in a formal language, that may be perceived by an agent, and hence offer an agent affordance. Consequently, the remainder of the discussion only considers explicit norms. In contrast to the view expressed in [8], which considers whether affordances are normative, the question here is aligned with [3], which also considers norms as affordances, in this case, for policy-making.

The concept of norm originates in the social sciences [7] to capture the conventions that guide behaviour in social interaction, while in economics [6] it captures formal and informal regulations, that inform agents how to act in a norm-compliant way, and the consequence of taking a particular action in certain circumstances, which may be a violation if an action is prohibited/not permitted, or an obligation to fulfil. [7] calls a collection of norms a social institution, others say policy or regulations.

Norms are an essential part of the social environment constructed between agents by reference to the social institution that governs the interaction, both in terms of the detached norms – different parties are aware of what others are allowed/obliged to do – but also in terms of which abstract norms may be employed to achieve their individual or mutual goals. One way of explaining and visualising this is given in Figure 5, where agents act and interact via the environment, in so doing each may construct some internal state, as a consequence of their perceptions of the environment, resulting in action choices, while the institutions interpret those actions against the set of norms each encapsulates, to provide social observations about those actions, which agents may also perceive, if they have the capability, and thus incorporate in their internal state to bring a normative dimension.

Whether a set of norms may constitute a social affordance largely depends on the agent's reasoning capability, but a capacity to reason about abstract norms, can inform a planning process for an agent to achieve some goal [1, 2]. Different agents may construct equivalent

[28] In case there exists perceptual information about an affordance, then the user is presented with a perceived affordance. The perceptual information may refer to a real affordance or a non-existent one, namely a false affordance [19].

[29] https://jnd.org/signifiers_not_affordances/

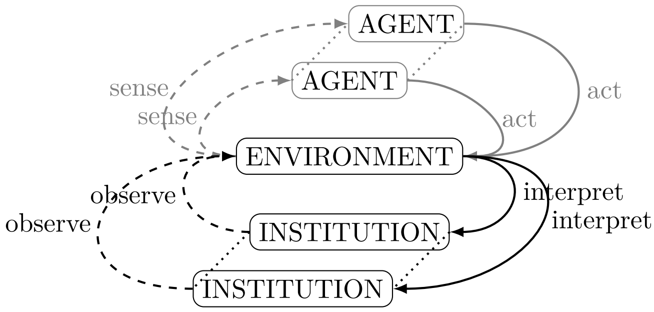


Figure 5 Institutions as normative sensors.

but different plans to fulfil the same goal, different plans to achieve different goals, or a mutual plan that delivers different outcomes, depending on participant role. In this way, although intangible, a social institution comprising explicit norms, which presents the same to all agents, may be perceived individually as offering quite different affordances.

6.2.3 Affordances on the Web

The term *hyptertext* was first coined by Ted Nelson in 1965 [13]. For Nelson, hypertext afforded the ability to connect multiple documents via selected text using a separate overlay file. Hall's MICROCOSM [14] was an early example of a hypermedia system that followed Nelson's principles. Berners-Lee built on Nelson's ideas to create what he called the World Wide Web [15]. In an important departure from Nelson and Hall, the WWW relied upon linking affordances embedded directly *within* documents themselves.

6.2.3.1 Affording the Web with Links and Forms

On the Web, affordances appear as links and forms. A link is typically used to navigate from one "place" in the Web to another "place". A form is used to collect and transmit data between the software agent (the client) and the web service (the server).

Successful software agents can recognize affordances within a particular language (HTML, Hydra[30], Collection+JSON[31], etc.) as well as activate them. For HTML agents, the task is to recognize and render the affordances as they appear. HTML browsers rely upon the human "driving" the agent to handle the work of activating them. Software agents not only need to recognize and parse the details of each affordance, they must also make decisions about which (if any) of the affordances could be used to achieve a particular outcome and, if needed, to populate and activate the selected affordance.

In the case of links, a navigation transition might actually be moving between specific locations within the same document or from one document to another. Sometimes the act of navigation does not represent relocation and, instead, represents a change in the state of the document (or environment).

In the HTML format, some common link affordances are the `a` (anchor) and `img` tags. Each affordance informs the HTML browser how to, for example, navigate between locations (`a`) or instructs it to fetch content from a remote location and embed it within the current document (`img`).

[30] `https://www.hydra-cg.com/spec/latest/core/`
[31] `http://amundsen.com/media-types/collection/`

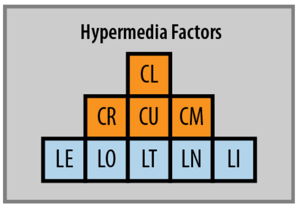


Figure 6 Hypermedia Factors.

In the case of forms, software agents are prompted to collect data and transmit it to another location on the Web. This transfer can have the effect of altering the state of the current document, modifiying a different document, or creating new documents. Essentially, form affordances are link affordances that also support data variables.

In the HTML format, the `form` tag is used to identify the basic elements of the affordance along with one or more `input` tags that each describe the data to collect and transmit.

On the Web, affordances can be characterized by a finite set of nine *hypermedia factors*[16] that may or may not be a part of an affordance's definition. These factors are explicit in the text markup of the affordance (for example, the `href` attribute of an HTML `a` tag) or implied (as in the valid binary image format types supported by the HTML `img` tag).

The nine factors are:

1. **LE – Embedding links**: Supports a read-only request (HTTP `GET`) and displaying the response within the current document (e.g. HTML `img` tag)
2. **LO – Outbound links**: A read-only request that treats the response as a navigation to a new document (e.g. HTML `a` tag).
3. **LT – Templated queries**: A read-only request like LO and LE with the ability to supply runtime parameters to vary the query (e.g. HTML `GET` form).
4. **LI – Idempotent updates**: Supports submitting data to the server in a way that is repeatable w/ the same results (e.g. HTTP `PUT`)
5. **LN – Non-Idempotent updates**: Supports submitting data to the server in a way that, when repeated, may return different results (e.g. HTTP `POST`) .
6. **CR – Control data for read requests**: Supplies metadata to modify the response (e.g. HTTP `ACCEPT-LANGUAGE` header).
7. **CU – Control data for update requests**: Supplies metadata data about the content of the request (e.g. HTTP `CONTENT-TYPE` header).
8. **CM – Control data for interface methods**: Supplies metadata about the protocol method to use for the request (e.g. HTML `form.method` property).
9. **CL – Control data for links**: Supplies metadata about the domain semantics of the affordance (e.g. HTML `link.rel` property).

Along with hypermedia factors, affordances on the Web can be further described using four *aspects* [17]:

(i) mutability: can the affordance be edited?
(ii) transclusion: is the response a navigation or an update to the current document?
(iii) idempotence: does the action result in the same response when repeated? and
(iv) safety: does the action modify the state of the remote machine?

The ability to recognize document affordances, categorize them by their hypermedia factors, and determine the status of the four hypermedia aspects represents a generalized resolution mechanism that can be programmed into software agents (see Sec. 6.2.4.1).

6.2.3.2 Affordances on the Web of Things

Activating a link or form affordance consists in sending a request to Web servers for manipulating information. The concept of affordance has been however defined by Gibson and Norman (Sec. 6.2.2.1) for manipulating physical objects. The Web of Things (WoT), as an architectural principle, bridges the gap between the information world and the physical world by exposing "things', i.e. physical objects, as Web resources that agents can manipulate in a RESTful manner. A model for describing "things" on the Web has been standardized by the W3C [10]. This model is centered on the concept of *interaction affordance*.

Interaction affordances for a particular "thing" are exposed on the Web as a Thing Description (TD), an RDF document that includes links and forms for agents to resolve. Listing 1 gives an example of a TD document for a lamp. The TD model includes three kinds of interaction affordances

1. to read or write a property of the "thing",
2. to invoke an action to be performed by the "thing" and
3. to subscribe to events monitored by the "thing".

The example TD presented here has an affordance of each type: one to read the `status` property of the lamp, another to invoke the lamp's `toggle` action and a last affordance to subscribe to an `overheating` event. All affordances include a Web form that agents can submit to (actively) observe or update the state of the lamp.

The main extension point of the TD model compared to other hypermedia formats (HTML, in the first place) is the classification of affordances with respect to a model of the physical world. The basic model for "things" assumed in a TD is derived from the Semantic Sensor Network (SSN) ontology, another W3C standard [11]. SSN models physical objects as "features of interest" that have physical properties.

Interaction affordances declared in a TD allow Web agents to interact with physical objects (modeled as SSN features of interest) by making observations and actuations on their properties. A Web form to rate online content on a social network or to place orders on an e-commerce platform should not be modeled as interaction affordances for they do not have a direct relation to any object's properties.

Agents perceive physical objects placed in a hypermedia environments only through interaction affordances. As per Gibson and Norman's definitions of affordances, interaction affordances are not themselves properties of physical objects. Rather, they are resources that Web agents consume, the only kind of agents capable of following links and submitting forms. By reducing the description of "things" to a description of their affordances to Web agents, the TD specification closely follows the principle of economical perception. It also lays the foundations of a new class of Web agents that can deal with complex cyber-physical systems. The main challenge when programming such agents is in properly selecting affordances and *resolving* signifiers found in the environment.

6.2.4 Signifiers

6.2.4.1 Signifier Resolution Mechanism

To be able to act upon signifiers that have been placed in the environment by environment designers or by other agents, agents make use of a mechanism that we refer to as *Signifier Resolution Mechanism* (SRM). Observing the principle of economical perception, an SRM is active whenever the agent perceives the environment; the SRM should also be executing relatively *fast*, compared to the agent's reasoning. Furthermore, the SRM acts upon an agent's

Listing 1 Taken from `https://www.w3.org/TR/wot-thing-description/`.

```
{
    "@context": "https://www.w3.org/2019/wot/td/v1",
    "id": "urn:dev:ops:32473-WoTLamp-1234",
    "title": "MyLampThing",
    ...
    "properties": {
        "status" : {
            "type": "string",
            "forms": [{"href": "https://mylamp.example.com/status"}]
        }
    },
    "actions": {
        "toggle" : {
            "forms": [{"href": "https://mylamp.example.com/toggle"}]
        }
    },
    "events":{
        "overheating":{
            "data": {"type": "string"},
            "forms": [{
                "href": "https://mylamp.example.com/oh",
                "subprotocol": "longpoll"
            }]
        }
    }
}
```

percepts of the environment, i.e., after information from the environment has undergone a possibly agent-specific perception mechanism (human analogy: color-blindness; software agent analogy: ability to parse different media types).

An SRM produces as output an ordered list of possibilities of how the agent could use a capabilities of objects in the percept, where each possibility is coupled with a motivation level (or priority) of acting upon it. If the SRM is related to a signifier of a thing, then the output is a (more or less strongly motivated) possibility of the agent to use a capability of the thing. Thus, an agent can perceive a signifier and resolve it to an affordance depending on the agent's percept of the thing's capabilities and state, and on the agent's abilities and the *strength* of resolved affordances depends on the agent's current internal state (including its goals) and the salience of the signifiers of the thing that resolve to the affordance.

To produce this ordered list, the SRM takes as input:

- A *percept* of the agent. A human agent might perceive a switch and its state of being on. A software agent might perceive a (machine-readable) description of the switch (e.g., a Thing Description) that includes triples such as `"MyThing" td:hasActionAffordance "toggle"`.
- The *abilities* of the agent. This is required for determining whether an agent's abilities match the capabilities of a Thing. A human agent does not resolve a tiny switch that the agent cannot operate physically to an *toggleable* affordance. A software agent does not resolve a signifier that it cannot parse syntactically to an affordance.
- The *internal state* of the agent (including the agent's goals). This is required for economical perception through signifiers and is the main information that is used to compute a ranking of (resolved) affordances. For agents, even if a switch affords toggling, the agent might not be motivated to toggle it because this conflicts with its goals.

After resolving affordances in its environment, an agent can decide to follow the highest-ranked affordance, or it can explicitly reason about the affordances and about which one it should follow.

If the agent finds itself closer to its goal after acting upon the affordance, then the resolution process was helpful (i.e., useful with respect to this agent, given the situation of this agent) and the affordance was a true affordance that the agent might decide to reinforce, e.g. through stigmergy (see Sec. 6.2.4.2).

6.2.4.2 Stigmergy

As mentioned in the previous section, agents can place signifiers in their environment. Depending on how the environment is designed, it might also be possible for agents to delete or alter existing signifiers (e.g. to increase their salience) or signifiers could be influenced by a process in the environment (e.g. evaporation). The signifiers placed in the environment can be used by other agents and should help them (either as input to the SRMs they already have or as a new SRM they can execute) to choose the right affordances to reach their goals.

This indirect coordination mechanism among multiple agents through their environment is called *stigmergy*, a term first coined by the French zoologist Pierre-Paul Grassé in 1959 to describe the behavior of social insects [12].

Agents that are situated in a hypermedia environment[32] can crawl the hypermedia graph and thus in principle perceive every signifier that is located at any reachable location. However, it would be very inefficient if hypermedia agents had to crawl their whole enviroment and resolve all signifiers every time they want to make an expedient decision. Through stigmergy, the information that is available globally (by crawling the hypermedia graph) can be made available locally: agents place signifiers only at the very locations where they are useful for other agents, per reinforcement. Simple reflex agents satisfying the principle of economical perception are then able to make decisions fast, though SRM only.

Looking at the example from Listing 1 we could imagine an agent that wants to switch on the light but does not resolve the `toggle` affordance, despite its ability to submit forms. This might happen if the agent has no internal representation for the “toggle”, as a concept. Other agents can now help our agent by either placing an additional signifier (i. e. it would change the TD to include another Action Affordance *switch_on* that has the same form as *toggle*).

In both cases it is most useful for our agent if the signifier is placed directly within the TD. It would, in principle, also be perceivable for the agent if it were placed at `http://purl.org/dc/terms/license` (as this resource is indirectly reachable from our TD), however this would need our agent to dereference every URI in the TD and resolve all the signifiers, which would be very inefficient.

6.2.5 Acknowledgements

We would very much like to thank Alessandro Ricci and Samuele Burattini who attended the Affordances and Signifiers working group and contributed to the discussion.

References

1 Zohreh Shams, Marina De Vos, Julian Padget and Wamberto W. Vasconcelos (2017). *Practical reasoning with norms for autonomous software agents.* In Engineering Applications of Artificial Intelligence 65, pp. 388–399. DOI:https://doi.org/10.1016/j.engappai.2017.07.021

[32] Hypermedia agents are situated insofar as the set of links and forms they can manipulate at a given time is finite.

2 Zohreh Shams, Marina De Vos, Nir Oren and Julian Padget (2020). *Argumentation-based reasoning about plans, maintenance goals, and norms.* In ACM Trans.Auton. Adapt. Syst.14.3, 9:1–9:39. DOI:10.1145/3364220

3 Pablo Noriega, Jordi Sabater-Mir, Harko Verhagen, Julian Padget and Mark d'Inverno (2017). *Identifying affordances for modelling second-order emergent phenomena with the wit framework.* In Autonomous Agents and Multiagent Systems -AAMAS 2017 Workshops, Visionary Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers. Ed. by Gita Sukthankar and Juan A. Rodriguez-Aguilar.Vol. 10643. Lecture Notes in Computer Science. Springer, pp. 208–227. DOI:10.1007/978-3-319-71679-4_14

4 Gita Sukthankar and Juan A. Rodriguez-Aguilar, eds (2017). *Autonomous agents and multiagent systems – AAMAS 2017 Workshops, Visionary Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers.* Vol. 10643. Lecture Notes in Computer Science. Springer. DOI:10.1007/978-3-319-71679-4

5 Thomas C. King, Marina De Vos, Virginia Dignum, Catholijn M. Jonker, Tingting Li, Julian Padget, and M. Birna van Riemsdijk (2017). *Automated multi-level governance compliance checking.* In Autonomous Agents and Multi-Agent Systems,pp.1–61. DOI:10.1007/s10458-017-9363-y.

6 Douglass North (1991). *Institutions, institutional change and economic performance.* CUP

7 Elinor Ostrom (1990). *Governing the commons. The evolution of institutions for collective action.* CUP

8 Manuel Heras-Escribano and Manuel de Pinedo (2016). *Are affordances normative?.* In Phenomenology and the Cognitive Sciences 15, pp. 565–589. DOI:https://doi.org/10.1007/s11097-015-9440-0

9 James J. Gibson (1979). *The ecological approach to visual perception: classic edition.* Psychology Press. DOI:10.4324/9781315740218

10 Sebastian Kaebisch, Kamiya Takuki, Michael McCool, Victor Charpenay and Matthias Kovatsch (Apr. 9, 2020). *Web of things (wot) thing description.* W3C Recommendation

11 Victor Charpenay and Sebastian Kaebisch (2020). *On modeling the physical world as a collection of things: the w3c thing description ontology.* In The Semantic Web. Ed. by Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez. Vol. 12123. Series Title: Lecture Notes in Computer Science. Cham: Springer Inter-national Publishing, pp. 599–615.DOI:10.1007/978-3-030-49461-2_35

12 Guy Theraulaz and Eric Bonabeau (1999). *A brief history of stigmergy.* In Artificial life 5.2, pp. 97–116

13 Theodor H. Nelson (1965). *Complex information processing: A file structure for the complex, the changing and the indeterminate.* In Proceedings of the 1965 20th National Conference. DOI:10.1145/800197.806036

14 Andrew M. Fountain, Wendy Hall, Ian Heath and Hugh C. Davis (1990). *Microcosm: An open model for hypermedia with dynamic linking.* In ECHT. Vol. 90, pp. 298-311. DOI:10.5555/129655.129677

15 Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik F. Nielsen, and Arthur Secret (1994). *The world-wide web.* Communications of the ACM 37, pp. 76-82. DOI:10.1145/179606.179671

16 Mike Amundsen (2011). *Hypermedia types.* In REST: From Research to Practice. Springer, New York, NY, 2011. pp. 93-116. DOI:10.1007/978-1-4419-8303-9_4

17 Mike Amundsen (2011) *Hypermedia-oriented design: An approach for supporting evolvable distributed network applications*

18 Anthony Chemero (2003). *Radical empiricism through the ages.* Contemporary Psychology 48 pp. 18–20

19 William W. Gaver (1991). *Technology affordances.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems
20 Thomas A. Stoffregen (2000). *Affordances and events.* Ecological psychology 12.1 pp. 1-28
21 William H. Warren (1984). *Perceiving affordances: Visual guidance of stair climbing.* Journal of Experimental Psychology: Human Perception and Performance 10, pp. 683-703
22 Joanna McGrenere and Wayne Ho (2000). *Affordances: Clarifying and evolving a concept.* Graphics Interface. Vol. 2000, pp. 179-186
23 James G. Greeno (1994). *Gibson's affordances*
24 Don Norman (2013). *The design of everyday things: Revised and expanded edition.* Basic books

6.3 Interaction Protocols

Samuel Christie (Lancaster University, GB), Jean-Paul Calbimonte (HES-SO Valais – Sierre, CH), Ege Korkan (TU München, DE), Maxime Lefrançois (Mines Saint-Étienne, FR), Simon Mayer (Universität St. Gallen, CH), and Munindar P. Singh (North Carolina State University – Raleigh, US)

6.3.1 Introduction

The multiagent systems (MAS) community studies MAS in terms of abstractions such as agents, organizations, norms, and protocols both to better model and understand human societies, and to design similar systems in which autonomous agents can cooperate. Interaction protocols are one approach to specifying MAS, focusing on the information or messages communicated between agents.

The MAS perspective on interaction protocols is relatively unknown in the Web and Semantic Web communities, so our objective as a workgroup was to understand the insights of the MAS community, and relate them to current work and research questions in the Web and Semantic Web spaces.

Our understanding of MAS and protocols was mostly derived from work on BSPL, the Blindingly Simple Protocol Language [24], partly because it is a primary research interest for two participants (Samuel Christie and Munindar Singh), and partly because its simplicity and information focus match well with ideas from Web and Semantic Web. However, we also invited Jomi Hubner to help us understand the perspectives of other parts of the MAS community, specifically as relates to the JaCaMo agent framework, and the Agents and Artifacts metamodel (A&A).

6.3.2 Background

6.3.2.1 Interaction Protocols

A protocol is an interaction specification; it declares constraints on what information each agent may send, and under what conditions it may be sent.

BSPL specifies a protocol as an interaction between two or more roles, consisting of one or more messages. Roles are abstract, and can be played by any capable and willing agent. BSPL takes an information-based approach to protocol specification, instead of a control-flow-based approach. The messages contain information parameters, and are constrained by causality (parameters may be dependencies, preventing an emission until known) and integrity (parameters may not have multiple bindings in an enactment). In respect for autonomy, constraints identify when a message is enabled; the agent may choose whether and when to send the message.

One example protocol we considered was the following Purchase protocol:

Listing 2 Purchase Protocol in BSPL.

```
Purchase {
  roles Buyer, Seller
  parameterS out ID key, out item, out shipment
  private price, payment

  Buyer -> Seller: RFQ[out ID key, out item]
  Seller -> Buyer: Quote[in ID key, in item, out price]
  Buyer -> Seller: Pay[in ID key, in price, out payment]
  Seller -> Buyer : Ship[in ID key, in payment, out shipment]
}
```

In this protocol, enactments are identified by the key `ID`; for each binding of `ID`, each other parameter may also only be bound once. The keywords `out` and `in` identify the dependency relationships; `out` binds the parameter when sent, and `in` means the message cannot be sent without observing that parameter in another message. Thus, Buyer can't pay until Seller specifies the price.

6.3.2.2 Other MAS Concepts

BSPL is focused on interactions from a local information perspective; each agent makes decisions only on the information they observe, and all information is explicitly communicated via messages. This model is intentional, to support autonomy and asynchrony by keeping all interactions between agents at a distance. Messages map well to stateless interactions using Web protocols such as HTTP.

However, other parts of the MAS community follow contrary models that focus on situated agents, placed in an environment and surrounded by artifacts. In this model, artifacts can be directly observed or invoked by an agent, and are the basis of implicit communication between agents in a shared environment. This model fits real-world settings, such as robotics.

6.3.2.3 Web and the Linked Data principles

The Web is the information space built on top of the Internet. Objects of interest on the Web are called resources, and are *identified* by IRIs. What goes through the wire are *representations* of resources, which are octet streams, typically typed using mediatypes. The Web architecture principles include the notion of *Web's protocols*, which define how one can access resources, the semantics of "access" being defined by the Web's protocols themselves, for example the HTTP request verbs GET, POST, PUT, etc.

The Web of data is a vision where a huge amount of data is made available on the Web in standard, machine-readable formats. Hypermedia relations among data are also made available, leading to the so-called Web of Linked data. The first principle of the Linked Data is that resources, and relations between them, are identified by IRIs.

With the rise of the Internet of Things (IoT), a huge number of connected devices are on the verge of entering the Web, use its architecture principles, and form the Web of Things. The W3C Web of Things (WoT) Working Group aims at developing standards to allow for the description of Things on the Web of Linked Data in terms of their affordances (property, action, and event affordances), and hypermedia forms to trigger those affordances. Protocol bindings are used to map affordance descriptions to lower-level Web or IoT protocol operations.

6.3.3 Summary of Results

6.3.3.1 Mapping to Web (of Things)

One important outcome of the discussions in the working group was deciding on how to relate Interaction Protocols to the concepts that exist in the Web (of Things) and compare to them. The concepts of affordance and hypermedia based design were the two prominent ones.

Comparison of an Affordance with an Interaction Protocol. An affordance can be thought as a function a web agent can invoke in order to do a single operation with the *owner* of the affordance, e.g., to read or change the value of a property, to trigger an action to start/stop a function, or to subscribe to a stream of events (these are mapped to property, action, and event, affordances, respectively). In contrast, an interaction protocol is a specification of an interaction between multiple agents; protocols are enacted instead of executed. Making a purchase (involving multiple actions by buyer and seller) could be specified as an interaction protocol, whereas the single action of paying for the product could be an affordance. A chain of affordance executions is not necessarily a protocol.

Comparison with Hypermedia. Hypermedia driven environments usually have well-defined endpoints that a client can use to execute an operation, which then provide the possible follow-up operations in their response. It means that a client does not have to know in advance what the next steps would be, allowing flexible implementations that adapt to the changes in the IRIs, methods, etc. Interaction protocols are usually prespecified and less dynamic, stating all of the possible actions up front instead of progressively revealing them. Theoretically, an hypermedia endpoint can be used as an entry point to announce an interaction protocol specification, but to the best of our knowledge this has not been done before.

6.3.3.2 Specifying different levels of Interaction Protocols

To understand interaction protocols, it is helpful to distinguish different levels of protocols illustrated by examples. The following levels are given in order from least to most flexible:

1. A trace of an interaction (over a group of messages): This would not be a protocol but an instance of it which can be seen analogous to a Schema and its respective Data. **Implementation:** Developers need to write code that needs to follow exact execution of certain operations, no loops, no branches, no states to manage.
2. Standard-like protocols: Protocols like OAuth (a certain flow), OCF Cloud Onboarding or even HTTP as protocol on top of TCP, would be in this category. These offer one way to do something; sometimes there can be configuration parameters but they are not application specific. **Implementation:** Developers would mostly use external libraries that execute these protocols according to some configuration parameters.
3. Interaction Protocols (BSPL): These specify the protocol by detailing only the constraints and the possible messages under these constraints. BSPL-based protocols detailed in Section 6.3.2.1 would be examples of these. **Implementation:** Developers can manually implement an agent or use a protocol adapter library to validate messages against their schemas and invoke appropriate handlers according to the protocol specification.
4. Hypermedia: A single endpoint would guide the Agent/Consumer and the protocol would be self-exploratory. **Implementation:** The endpoint would choose a standard that describes the payloads/message. The payloads/message can be understood to form decisions by the developers who program the agents.

6.3.3.3 Interaction Protocol Templating in TDs

Where interaction protocol specifications are stored or how they are exchanged is not a major focus of the multiagent systems community. One solution could be to use WoT Thing Descriptions to specify which protocols a Thing can participate in.

The BSPL protocol in Listing 3 describes a temperature reading scenario, where a reader (a WoT Consumer) can read temperature values from a temperature sensor. We have worked on a draft TD document for the Sensor's role in this protocol, given in the Appendix in Listing 4. We expect that this first attempt can be refined to improve expressiveness and integration with TD documents.

Listing 3 Temperature Reading Protocol in BSPL which is relatable to WoT operations.

```
Temperature Reading {
  roles Sensor, Reader, Receiver
  parameters out ID key, out temperature_value

  out Reader -> out Sensor: RFT[out ID key, out Receiver]
  Sensor -> in Receiver: Temperature[in ID key, out temperature]
}
```

6.3.4 Participant Comments

6.3.4.1 Samuel H. Christie V

My research chiefly concerns information-based interaction protocols, and I ended up leading most of the discussion around that part of the topic. Some of the work that I have done with my advisor (Munindar Singh) on the subject includes defining atomicity for protocols [8], defining refinement for protocols to slightly modularize verification [9], applying protocols to an IoT logictics scenario [10], and beginning work on a concept of application-level fault tolerance, using a protocol specification to derive agent expectations and thereby detect and recover from faults [11].

Protocols and the Web seem very well suited to each other. Linked data can be used to specify types and meanings of message parameters. Protocols can be used as an alternative to APIs for specifying interfaces and interactions; hence adding some protocol concepts to Thing Descriptions is an interesting direction. Finally, our work on protocols has not really addressed the negotiation and initiation phases of protocol enactment; using the Web to offer and consume protocol specifications seems like a promising approach.

I believe both areas can greatly benefit from this collaboration; the Web because protocols help address difficulties regarding multiagent interactions and decentralized systems to help move beyond client-server, and protocols because of the attention and refinement that industry exposure and application bring.

6.3.4.2 Jean-Paul Calbimonte

Background. Interaction protocols are fundamental in order to establish principled communication among agents on the Web. In the same way specifications have been provided for agreeing on ontologies and semantic data models, it should be possible to have a formalization of the way agents interact and exchange messages, or express their goals and future behaviors. Our previous work have been focused on bridging multi-agent systems and semantic Web, especially regarding stream processing and dynamic data (e.g., provided by sensors and IoT devices). A first approach in this direction was to use Linked Data Notifications (LDN) as the basis for orchestrating interactions among agents on the Web [3]. In the more specific context of eHealth, we also explored the use of Web agents to manage patient trajectories [4], which also led to orthogonal research regarding privacy ontologies for personal data handled by multi-agent systems [5]. Some of these challenges, which touch the question of how interaction protocols could/should be specified, were also considered in a blue-sky paper where we introduced the concept of stream reasoning agents [30].

Challenges. One of the first questions raised around this issue is why do we need interaction protocols in the first place. In the context of the Web, discovery of services and capabilities has been studied in depth, for instance in the context of Semantic Web Services. Regardless of the degree of adoption of these specifications, they may serve as a source of inspiration for future work on interaction protocols. This is linked to the description of affordances (e.g. in Things Descriptions) but also to the specification of how agents on the Web can initiate a negotiation, cooperation or joint-decision-making process.

Another key challenge is related to the heterogeneity of agent interactions, including the different offerings that they provide. Although agent beliefs can be represented in a general sense by knowledge graphs and using ontologies and other elements form the Semantic Web, it still remains to be seen how this applies also to agent behaviors.

It is also fundamental to study how the social aspects of agent interactions play a role in the specification of protocols. Agents do have a large degree of autonomy, but this does not exclude the possibility of engaging in social coordination or in a collaboration scheme. Interaction protocols may specify what an agent may offer or gain in such a negotiation process.

Considering that many of the discussed scenarios are linked to IoT and sensing scenarios, the challenge of handling sensitive data should also be addressed, especially concerning privacy and ethics. Agents might be able to handle knowledge graphs including personal data, which require incorporating privacy preserving mechanisms and compliance to legal standards such as GDPR.

Future directions. Addressing these challenges, several directions can be explored, including:

- Specification of interaction protocols in conjunction with existing standards in IoT and WoT such as Thing Description.
- Exploration of specification of coordination and negotiation protocols for agents on the Web.
- Formalization and standardization of reasoning goals and expectation in Web Agents' interactions
- Study and specification of social interactions among Web Agents.
- Study ethics and accountability on Web Agent interaction protocols.
- Study of applications in different use cases, e.g. eHealth, manufacturing, robotics, industry 4.0, etc.

6.3.4.3 Ege Korkan

Background. I am a Ph.D. student and researcher on the Web of Things (WoT). My research is heavily based on the assumption of standardized WoT, where each Thing is represented by a Thing Description (TD). On top of that, I try to build well defined approaches on how to develop systems, i.e. describe systems, implement and then test them. On this front, related to the Interactions Protocols, I have worked on describing possible execution paths for WoT Consumers [17] and also full WoT Systems [18].

When approaching MAS, I always have the notion of Thing being entities that offer a service that is mostly one-sided, i.e. a Consumer can read a temperature, can turn robot to the left, get notifications of a button pressed. In contrast to interaction protocols that I have learned of during the seminar, WoT interactions require no coordinated effort by the two parties.

Highlights. From my point of view, the highlights of the work we have done in this short time were understanding different types/levels of (interaction) protocols, their representations and possible representations in TD-like documents for the WoT. Identifying the different levels of interaction protocols is fundamentally important since they are structured examples that allow others to orient their arguments. These levels were documented in the Section 6.3.3

The other highlight is what we have started doing on the last day, the interaction protocol templates in TD-like documents. Things would specify protocols they can participate in by giving a set of parameters and constraints. An example of this can be found in Appendix 1.

Challenges. Since the seminar is about bringing different communities together, I think that one challenge is to convince the others on why the collaboration is necessary. This can be achieved by putting examples that show that somethings can be done easily/elegantly in one community and difficultly in other ones. An example for the WoT:

- Advantages:
  - Separation of concerns: The Thing offers services and they are independent of other Consumers. No need to use a framework that couples them
  - Ease of Implementation: No need for specific libraries, any code that can send a request with a certain protocol (HTTP, MQTT, Modbus, etc.) can be used.
  - Brownfield Compatible: One can take an already existing device and make it WoT-compatible. Examples are Philips Hue devices
- Disadvantages:
  - Difficult to offer complex interactions: Since each interaction is seen to be stateless, how does a Consumer how to operate a Thing that requires multiple message sequences.
  - Lack of autonomous behavior: Since each TD can be different and offer different endpoints and may not use semantic annotations, it is simply not possible for vanilla WoT Things and Consumers to participate in autonomous activities.

A further example for MAS (almost the contrary of the above):

- Advantages:
  - Abundance of methods for autonomous systems: Since every entity can be properly modeled and implemented, it makes it easy to forget about the payload formats, communication protocols etc. and focus purely on the application logic, which can be then made autonomous.
- Disadvantages:
  - Closed Systems: Agents are part of a system or a framework that is tightly coupled, making it difficult to introduce new agents that are not part of the framework.

Research Directions. I believe that the interaction protocol templates ideas that we have started in the last day of the seminar is an important research direction to pursue. It allows complex interactions to be described for WoT-only communities but at the same time makes it possible to integrate WoT Things to MAS environments. Thus, one can parse such a template together with its TD to integrate them to such an environment.

6.3.4.4 Maxime Lefrançois

I am interested in the use of Semantic Web technologies for achieving Semantic Interoperability on the Web, and also on the Web of Things given all the additional challenges it brings in the picture [1, 6]. I acknowledge that the RDF data formats will never replace the plethora of datatypes that are used for representing resources on the Web. One of my lines of research consists in investigating how one can still consider that the content of information resources is modeled with the RDF data *model*. This sheds new light on the various data formats, and how they can become part of the Web of Linked Data at minimal cost [19, 2].

When asked about how to convert some piece of data or some data format specification to the Semantic Web, I often insist on two very different end goals which are: reasoning with the data, and distributing and linking the data on the Web. The latter goal, although much less salient, usually helps to cover many useful use cases, and rises interesting research questions.

For example, interesting research questions arise when discussing how BSPL specifications could be distributed and linked on the Web. This brings the question of commonly identifying messages or parameters in different BSPL documents to operate the join. One essential step that would be required is to adopt the Web principles for identifying resources, that is: with IRIs.

Thing description provides an interesting level of abstraction on top of various internet protocols, which then become transport protocols for WoT interactions. The notion of forms with protocol bindings and operation types could be leveraged by BSPL to spread on the Web, leveraging the ever-growing list of media types and interaction protocols. Operation types in the TD could allow for identifiers for interaction protocols, for example some identifier for the "Quote" message in Listing 1, or an identifier of the AgentSpeak **Tell** speech act. The BSPL message parameters could also be specified further with a datatype, or be complex structures described in terms of their schema, much like JSONSchema for JSON documents or SHACL for RDF graphs.

Other vocabularies than TD exist to describe devices and how agents can interact with them, such as SAREF [14] or SOSA/SSN [16]. It could seem natural to use a dedicated RDF query or scripting language to enact these operations, for example as an extension of LD-Script [13] or SPARQL-Generate [20]. What additional effort would it require to use these vocabularies to describe agents with their possible roles in protocols, or messages they can receive or send ?

Many more synergies can be identified when gathering the multi-agent system, and the WoT communities. For example Thing Descriptions can be used to automatically generate code for artifacts in CArtAgO – a virtual environment framework for multi-agent systems [12].

6.3.4.5 Simon Mayer

I was only briefly part of this working group. One observation I had about a possibly valuable integration point between the MAS-oriented perspective and the Web/SemWeb-oriented perspectives is regarding the management of interaction protocols. I believe that it would be fruitful to use Web approaches – in particular HATEOAS-based guidance from a single entry point – to enable the negotiation and, possibly, also the parameterization, of (very structured) MAS interaction protocols. In this way, we could create systems that flexibly enter into *tactical* interactions between agents, and where the entry points into these interactions (i.e., interaction protocol templates) are expressed as semantically annotated

Web resources. Software agents could then discover these entry points when navigating hypermedia, parameterize them at run time, enter into tactical interactions with other agents, and continue navigating after one such tactical interaction has concluded. Building on this idea, we could see API bindings of W3C Thing Descriptions as (extremely) simple interaction protocols between agents and things; which would offer a bridge into the integration of agent/service and agent/agent interactions (and into stigmergy as well, see the *Affordances and Signifiers* working group).

6.3.4.6 Munindar P. Singh

I am interested in modeling and implementing decentralized systems in terms of the interactions among their member agents. Specifically, I think of interaction as a *first-class* concept in that we would model interactions directly, not in terms of behaviors of agents [27]. This thinking leads us to protocols as specifications of possible interactions between roles (abstracting from agents). Each protocol describes a multiagent system abstractly; a concrete system would include agents playing the roles in the protocol and providing the reasoning necessary to decide whether and how they would participate in the protocol.

That is, protocol is our unit of modeling and enactment. In the spirit of a formal model of protocol, we need to provide ways to refine protocols (e.g., a purchase protocol may be refined into a protocol for purchase by credit card) [9, 15, 21]. Likewise, we need ways to compose protocols (e.g., a purchase protocol may be composed from protocols for price discovery, order placement, shipping, and payment) [26, 29].

One challenge with protocols concerns an ability to enact them in a decentralized manner, i.e., without a central entity to coordinate the interactions of the participants. Earlier approaches sought to achieve such enactments based on temporal logic specifications of constraints on events [22, 23]. But conceiving of interactions as first-class abstractions provides for a clearer representation in which we model causal relations between interactions (not agents, although that is usually implicit) based on the information flow between interactions [24]. This approach makes an information-driven approach possible in which an agent can send information to another (via a message, viewed as an elementary interaction) [25].

Another challenge with protocols is to associate application-level meanings with them. Meanings provide a principled basis for agents to enact a protocol flexibly while complying with its specification at the level of meanings [7]. There is a tension between the meanings-based and operational (concerning message occurrence and ordering) specifications of interactions. Procedural representations for operations make it difficult to capture modularly. The above-mentioned information representation streamlines the operational specification and makes capturing meaning easier. Recent work shows how to map from a meaning-based specification to protocols [28].

Even as the above challenges are being addressed, some major challenges remain. First, achieving flexibility in protocols comes at the price of high computational complexity of verification algorithms. Second, we need new programming models for flexible protocols. Third, we need to investigate how to better meld protocols with web systems. Fourth, a study of edge computing from the standpoint of protocols would be interesting.

6.3.5 Acknowledgements

We would like to thank to Dagstuhl for hosting our seminar, and helping us to have a fun and productive time despite needing to meet remotely.

We thank the seminar organizers – Olivier Boissier, Andrei Ciortea, Andreas Harth, and Alessandro Ricci – for bringing us together and guiding these discussions.

We would also like to thank the following additional participants who attended our workgroup and contributed to the discussion: Andreas Harth, Jomi Hubner, and Danai Vachtsevanou.

Christie thanks the EPSRC (grant EP/N027965/1) and the US NSF (grant IIS-1908374) for support. Singh thanks the US NSF (grant IIS-1908374) for support.

References

1 F. Antoniazzi, G. A. Atemezing, F. Badeig, M. Bennara, S. Bernard, P.-A. Champin, J.-P. Chanet, C. Gravier, Y. Gripay, F. Laforest, M. Lefrançois, L. Médini, L. Moiroux, C. Roussey, S. Servigne, K. Singh, J. Subercaze, and A. Zimmermann. Interoperabilité et raisonnement dans le web sémantique des objets: le projet coswot. In *31es Journées francophones d'Ingénierie des Connaissances*, 2020.

2 M. Bennara, A. Zimmermann, M. Lefrançois, and N. Messalti. Interoperability of semantically-enabled web services on the wot: Challenges and prospects. In *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services*, pages 149–153, 2020.

3 J.-P. Calbimonte, D. Calvaresi, and M. Schumacher. Multi-agent interactions on the web through linked data notifications. In *Proceedings of the 15th European Conference on Multi-Agent Systems EUMAS 2017*, volume vol 10767 of *Lecture Notes in Computer Science*, pages 44–53, Evry, France, Dec. 2017. Springer. doi: https://doi.org/10.1007/978-3-030-01713-2_4.

4 D. Calvaresi, M. Schumacher, and J.-P. Calbimonte. Agent-based modeling for ontology-driven analysis of patient trajectories. *Journal of Medical Systems*, 44:158, 2020a. doi: https://doi.org/10.1007/s10916-020-01620-8.

5 D. Calvaresi, M. Schumacher, and J.-P. Calbimonte. Personal data privacy semantics in multi-agent systems interactions. In *Proceedings of 18th International Conference on Practical Applications of Agents and Multi-Agent Systems*, volume vol 12092 of *LNCS*, pages 55–67. Springer, June 2020b. Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection. PAAMS 2020.

6 F. Cena, A. Haller, and M. Lefrancçois. Semantics in the edge: Sensors and actuators in the Web of linked data and things, editorial. *Semantic Web Journal*, 11(4):571–580, 2020.

7 A. Chopra and M. P. Singh. Nonmonotonic commitment machines. In F. Dignum, editor, *Advances in Agent Communication: Proceedings of the International Workshop on Agent Communication Languages (ACL 2003)*, number 2922 in Lecture Notes in Artificial Intelligence, pages 183–200, Melbourne, 2004. Springer. doi: 10.1007/978-3-540-24608-4_11.

8 S. H. Christie V, A. K. Chopra, and M. P. Singh. Compositional correctness in multiagent interactions. In *AAMAS-18*, pages 1159–1167, Stockholm, July 2018. IFAAMAS. doi: 10.5555/3237383.3237868.

9 S. H. Christie V, A. K. Chopra, and M. P. Singh. Refinement for multiagent information protocols. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 258–266, Auckland, May 2020. IFAAMAS. doi: 10.5555/3398761.3398796.

10 S. H. Christie V, D. Smirnova, A. K. Chopra, and M. P. Singh. Protocols over Things: A decentralized programming model for the Internet of Things. *IEEE Computer*, 53(12): 60–68, Dec. 2020. doi: 10.1109/MC.2020.3023887.

11 S. H. Christie V, A. K. Chopra, and M. P. Singh. Bungie: Improving fault tolerance via extensible application-level protocols. *IEEE Computer*, 54(5):1–8, May 2021. doi: 10.1109/MC.2021.3052147. In press.

12 E. Chung, M. Lefrançois, and O. Boissier. Increasing interoperability in the web of things using autonomous agents (position paper). In *18e Rencontres des Jeunes Chercheurs en Intelligence Artificielle 2020 (RJCIA)*, 2020.
13 O. Corby, C. Faron-Zucker, and F. Gandon. Ldscript: a linked data script language. In *International Semantic Web Conference*, pages 208–224. Springer, 2017.
14 L. Daniele, F. den Hartog, and J. Roes. Created in close interaction with the industry: the smart appliances reference (saref) ontology. In *International Workshop Formal Ontologies Meet Industries*, pages 100–112. Springer, 2015.
15 S. N. Gerard and M. P. Singh. Formalizing and verifying protocol refinements. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 42(2):21:1–21:27, Mar. 2013. doi: 10.1145/2438653.2438656. Appendix pages 1–7.
16 A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler. The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1):9–32, 2019.
17 A. Kast, E. Korkan, S. Kaebisch, and S. Steinhorst. Web of things system description for representation of mashups. In *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, 9 2020. doi: 10.1109/COINS49042.2020.9191677. https://s-steinhorst.github.io/PDF/2020-COINS-Web%20of%20Things%20System%20Description%20for%20Representation%20of%20Mashups.pdf.
18 E. Korkan, S. Kaebisch, M. Kovatsch, and S. Steinhorst. Safe interoperability for web of things devices and systems. In S. S. Tom J. Kazmierski and D. Grosse, editors, *Languages, Design Methods, and Tools for Electronic System Design – Selected Contributions from FDL 2018*, pages 47–69. Springer International Publishing, Cham, 2020. ISBN 978-3-030-31585-6. doi: 10.1007/978-3-030-31585-6_3. URL https://doi.org/10.1007/978-3-030-31585-6_3.
19 M. Lefrançois. RDF Presentation and Correct Content Conveyance for Legacy Services and the Web of Things. In *8th International Conference on the Internet of Things (IOT'18)*, Santa-Barbara, United States, Oct. 2018. doi: 10.1145/3277593.3277618.
20 M. Lefrançois, A. Zimmermann, and N. Bakerally. A sparql extension for generating rdf from heterogeneous formats. In *European Semantic Web Conference*, pages 35–50. Springer, 2017.
21 A. U. Mallya and M. P. Singh. An algebra for commitment protocols. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 14(2):143–163, Apr. 2007. doi: 10.1007/s10458-006-7232-1.
22 M. P. Singh. A customizable coordination service for autonomous agents. In *Intelligent Agents IV: Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL-97)*, number 1365 in Lecture Notes in Computer Science, pages 93–106, Providence, Rhode Island, 1998. Springer. doi: 10.1007/BFb0026752.
23 M. P. Singh. Distributed enactment of multiagent workflows: Temporal logic for web service composition. In *Proceedings of the 2nd International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 907–914, Melbourne, July 2003. ACM Press. doi: 10.1145/860575.860721.
24 M. P. Singh. Information-driven interaction-oriented programming: BSPL, the Blindingly Simple Protocol Language. In *Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 491–498, Taipei, May 2011a. IFAAMAS. doi: 10.5555/2031678.2031687.
25 M. P. Singh. LoST: Local State Transfer – An architectural style for the distributed enactment of business protocols. In *Proceedings of the 9th IEEE International Conference*

on Web Services (ICWS), pages 57–64, Washington, DC, July 2011b. IEEE Computer Society. doi: 10.1109/ICWS.2011.48.

26 M. P. Singh. Semantics and verification of information-based protocols. In *Proceedings of the 11th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1149–1156, Valencia, Spain, June 2012. IFAAMAS. doi: 10.5555/2343776.2343861.

27 M. P. Singh and A. K. Chopra. Programming multiagent systems without programming agents. In *Proceedings of the 7th International Workshop on Programming Multiagent Systems (ProMAS 2009)*, number 5919 in Lecture Notes in Artificial Intelligence, pages 1–14. Springer, 2010. doi: 10.1007/978-3-642-14843-9_1. Invited paper.

28 M. P. Singh and A. K. Chopra. Clouseau: Generating communication protocols from commitments. In *Proceedings of the 34th Conference on Artificial Intelligence (AAAI)*, pages 7244–7252, New York, Feb. 2020. AAAI Press. doi: 10.1609/aaai.v34i05.6215.

29 M. P. Singh, A. K. Chopra, and N. Desai. Commitment-based service-oriented architecture. *IEEE Computer*, 42(11):72–79, Nov. 2009. doi: 10.1109/MC.2009.347.

30 R. Tommasini, D. Calvaresi, and J.-P. Calbimonte. Stream reasoning agents. In *Proceeding of International Conference on Autonomous Agent and Multi-Agent Systems (AAMAS 2019)*, pages 1664–1680, Montreal QC, Canada, May 2019. ACM. `https://dl.acm.org/doi/abs/10.5555/3306127.3331894`.

6.3.6 Appendix

Listing 4 A WoT Thing Description idea that can describe the Temperature Reading protocol of Listing 3 by defining additional terms.

```
{
  "@context": ["http://www.w3.org/ns/td",{}],
  "id": "urn:dev:ops:32473-WoTLamp-1234",
  "title": "Sensor",
  "securityDefinitions": {
    "basic_sc": {
      "scheme": "basic",
      "in": "header"
    }
  },
  "security": "basic_sc",
  "actions": {
    "receive": {
      "forms": [
        {
          "href": "https://sensor.example.com/receive"
        }
      ]
    }
  },
  "protocols": {
    "sendable": {
      "temperature": {
        "keys": [
          "ID"
        ],
        "to": {
          "name": "Receiver",
          "type": "string",
          "const": "http://localhost:80/recipient",
          "@type": "RecipientType"
        },
        "in": [
          {
            "name": "ID",
            "type": "string",
            "@type": "InteractionIDType"
          }
        ],
        "out": [
          {
            "name": "temperature",
            "type": "float",
            "@type": "Temperature"
          }
        ]
      }
    },
    "receivable": {
      "RFT": {
        "keys": [
          "ID"
        ],
        "action": "receive",
        "from": {
          "name": "Receiver",
          "type": "string",
          "@type": "RecipientType"
        },
        "out": [
          {
            "name": "ID",
            "type": "string",
            "@type": "InteractionIDType"
          }
        ]
      }
    }
  }
}
```

6.4 Norms and Policies for Agents on the Web

Timotheus Kampik (University of Umeå, SE), Olivier Boissier (Ecole des Mines – St. Etienne, FR), Sabrina Kirrane (Wirtschaftsuniversität Wien, AT), Adnane Mansour (Mines Saint-Étienne, FR), Julian Padget (University of Bath, GB), Terry Payne (University of Liverpool, GB), Munindar P. Singh (North Carolina State University – Raleigh, US), Valentina Tamma (University of Liverpool, GB), and Antoine Zimmermann (Ecole des Mines – St. Etienne, FR)

The study of policies and norms has a long tradition in the Semantic Web & Linked Data and Multi-Agent Systems communities, with applications ranging from business processes, and legal reasoning, to information systems governance. Although representing and reasoning about norms is crucial in ensuring that autonomous agents act in a manner to satisfy stakeholder requirements, normative concepts have yet to be considered as first-class abstractions in Web-based software systems. The chapter motivates the practical need to apply research on policies and norms to autonomous agents on the Web, and highlights research challenges and opportunities at the intersection of the Semantic Web & Linked Data, Multi-Agent Systems, and Web Architecture & Web of Things communities.

6.4.1 Introduction

When envisioning autonomous agents on the Web, it is crucial to consider a *governance* perspective that defines how an agent should act in a given situation, including the normative consequences of its potential actions and how frameworks that govern groups of agents are to be designed, interoperate, and evolve. This perspective is of particular importance for Web-based contexts, where usage may cross different social and legal jurisdictions, and where there is no centralised control over the provenance of the different agents.

This chapter makes two main contributions. First, it motivates norms and policies for *autonomous agents on the Web* using a hypothetical use case (Section 6.4.2) and provides a concise overview of the state of the art on norms and policies for autonomous agents and web-based systems to underline the need for alignment and joint research across communities (Section 6.4.3). Second, the research challenges and opportunities are identified for the Semantic Web & Linked Data, Multi-Agent Systems (MAS), and Web Architecture & Web of Things (WoT) communities (Section 6.4.4).

6.4.2 Use Case

Consider a hospital that is responsible for COVID-19 vaccine administration, supported by a range of IT systems and sub-systems. Some of these sub-systems may be considered as *autonomous agents*, which are assigned different roles and work towards fulfilling different goals. One agent manages the access and inventory of different COVID-19 vaccine doses stored within a specialised freezer, via a robotic arm, and dispenses the vials to other lab equipment or healthcare staff when required. The freezer itself is digitally represented by a WoT *Thing* (based on the W3C Web of Things *thing description* [42]), which in turn defines its *properties*, *actions* (and action handlers), and *events* (and event handlers).

Individual patients are represented by *patient agents* that manage and provide access to the patient's medical data; such access is determined by negotiation with *collector agents*. These agents obtain information about each patient whose agent requests a vaccination appointment and verify the patient's eligibility for receiving the vaccine. Furthermore, collector agents have the capacity to solicit patients through dissemination channels on the Web. An *organiser* seeks to compile lists of eligible patients based on both the determined criteria and information provided by the collector agents.

For this scenario, there are a number of "permissions" that an agent acting on behalf of a medical practitioner may have to consider when being asked to withdraw a vaccine vial:

- The agent must *authenticate* itself as legitimately acting on behalf of a medical practitioner.
- The agent should also validate the association the medical practitioner has with a particular *organisation*, i.e., with the healthcare organisation that is in charge of administering the vaccine or any sub-organisation to which the vaccine administration has been delegated.
- Most likely, the medical practitioner has a specific organisational *role* that may or may not authorise their agent to withdraw the vaccine, potentially depending on the current *context*. For example, the practitioner must be assigned to the *vaccination staff* role, and their current context must be set to *on duty*.

The issues listed above merely deal with the "permission" to access a vaccine dose from the freezer. There are also implied "obligations":

- The medical practitioner that the agent represents must ensure the dose is not wasted, i.e. it must be administered to a valid patient (i.e. one in need of the vaccine that satisfies the current eligibility criteria) before the expiry date[33].
- This practitioner must respect the priority order for vaccine administration (for example, the old and vulnerable population must be vaccinated first), unless disrespecting the priority order implies wasting the dose.
- This practitioner must not allow friends and relatives to "jump the line" and violate the priority order for for vaccine administration (unless a valid exception exists; see below for details).

The practitioner may rely on IT systems (i.e. "agents", see above) to make decisions; these agents need to consider these obligations as constraints. Further complications may arise if any of the agents attempt to negotiate a relaxation of the obligations, either in anticipation of or after a (potential or factual) violation.

- A patient agent can try to negotiate an exception for a potential obligation violation (through argumentation, for example, where the patient agent believes that the data it has for a patient satisfies the eligibility criteria but the same conclusion is not held by the collector agent). In such cases, the practitioner may have to choose between either administering a vaccine dose where eligibility is uncertain, or letting the dose expire.
- On behalf of the organisation, another agent may respond by granting or denying the request for an exception, or even by updating the organisations norms to generally permit the request.

These issues merely relate to the permissions needed for the obligations implied by opening the fridge. Additional challenges arise when considering the complete sociotechnical system, including electronic health record access [43] and supply chain integration [66]. This underlines the need for systematic and scalable approaches for the governance of autonomous agents on the Web that interact with other agents or with physical devices.

[33] In a practical scenario, we would not expect 100% compliance with this constraint, but the number of excess (wasted) doses relative to administered doses does not exceed a specified threshold.

6.4.3 Background

6.4.3.1 Web Architecture and the Web of Things

Bringing agents to the Web requires more than simply exploiting Web protocols (HTTP [31]) and data formats (e.g., XML [10], RDF [18]); the communication infrastructure used by agents and MAS should also comply with an architectural style based on well-defined principles, such as REpresentational State Transfer (REST [30]) as instantiated in the Architecture of the World Wide Web [40]. The Web of Things is an attempt to take advantage of the Web architecture [51] in the context of the *Internet of Things*: "things", which can be physical objects or virtual entities such as Web services, become resources that can be acted upon or queried via APIs (WoT scripting API [49]). To ensure that such "things" can be used without human intervention, they have to be formally described. To this end, the W3C standardised the *Thing Description* [42] which is a specification that defines how to provide a JSON-LD representation of the affordances (properties or actions) of a "thing" via Web APIs. On top of this, *WoT Discovery* [16] provides a mechanism for automatically discovering "thing" descriptions without hard-coding the location of the descriptions. These standards provide more autonomy for agents that make use of "things" connected to the Internet via Web standards. However, in terms of norms and policies, these standards offer only limited support [62], and are currently described in a set of guidelines targeted at human developers rather than as declarative, machine-readable statements usable by autonomous agents.

6.4.3.2 Semantic Web and Linked Data

[5] originally envisioned a system in which intelligent agents act on behalf of humans. A key component of such a system is a policy language capable of capturing the goals and constraints under which the agents operate. During the early days of the Semantic Web, the development of general policy languages that leverage semantic technologies (such as KAoS [9], Rei [44] and Protune [8]) was an active area of research. Indeed, much of the early work on policies coincided with the active development of rule languages such as SWRL [38] and RIF [46]. The study by [48] provides a detailed survey of the various access control models, standards and policy languages, and the different access control enforcement strategies for RDF. In addition to access control, there has also been work on *usage* control in the form of licensing [11, 71, 33, 34, 35], and more recently, policy languages have been used as a means to represent regulatory constraints [60, 21].

Some of the early Semantic Web research addressed decentralisation through P2P architectures, where bespoke protocols were defined to support the decentralised management and exchange of knowledge and information amongst networks of agents [67]. Although peers typically adhered to those protocols they could satisfy, many such protocols, in addition to defining the modalities of interaction, often defined illegal moves and penalties for violations; thus becoming de-facto *policies* [50]. Furthermore, Semantic Web approaches often assumed autonomous and rational agents as first-class citizens; thus there has been a focus more on how to support knowledge sharing in a centralised or decentralised fashion [26, 41], and on the type of explicit knowledge needed to support interoperability [53, 70, 24, 36].

From the *service* perspective, the use of ontologically grounded annotations for web services, or those services provided by autonomous agents dates back to the early days of the Semantic Web [39, 28]. The formalisms used to provide an ontological grounding and description of different kinds of services included both F-Logic [47] as used by WSMO [27], and DAML-S [19] (based on the DARPA Agent Markup Language) which later evolved as OWL-S [55]). Other approaches to support service utilisation were developed using OWL, e.g. the OWL ontology for protocols, OWL-P [22], or federated service discovery mechanisms such as the semantically annotated version of UDDI [61]. Although such services could be provided by autonomous agents, their modelling assumed functional service calls (through the invocation of endpoints with relevant arguments) rather than through the type of speech acts more familiar to autonomous agents.

6.4.3.3 Multiagent Systems

The study of norms is a long-running and active line of research in the Multi-Agent Systems (MAS) community, as evidenced by Dagstuhl seminars [20, 3], and a handbook on the topic [13]. Normative MAS [6] are realised in a number of ways, which are characterised on the one hand by the reasoning capabilities of the agents themselves, and on the other by whether norms are implicit or explicit with the support of monitoring and enforcement mechanisms. Agent capabilities vis-á-vis norms typically fall into three categories:

(i) *norm unaware*, agents do what they want, but may be regimented by external agencies to enforce norm compliance;
(ii) *norm aware*, agents may choose to comply or not with norms, depending on the alignment of their goals with those norms, the penalties for non-compliance, and the likelihood of enforcement; and
(iii) *value aware*, where agents, in addition to being norm aware, may be able to participate in norm creation and norm revision by reasoning about the value supported (or not) by particular norms.

Implicit norms that reside in the agents themselves are expressed through agent behaviour, and are not otherwise externally discernible. Explicit or referenceable norms may have an abstract representation involving variables, and a grounded (detached) representation in an entity such as a contract [65], institution [58, 25, 23], or organisation [69, 7]. Agents that are norm or value aware should be able to:

(i) recognise the norms;
(ii) decide whether they want to follow them; and
(iii) adapt their behaviour according to the norms, if they decide to do so.

Compliance with normative systems depends on how individual agents can reason and adapt to norms both at design and at run time [63, 68, 52].

6.4.4 Challenges and Opportunities

In this Section we present a range of research challenges and opportunities related to normative agents on the Web. The four horizontal challenges that characterise the contributions of norm-based Multi Agent Systems for the Web are presented below (Figure 7) in addition to a further two orthogonal challenges that need to be tackled to some extent to address the horizontal ones.

- **Norms and Interaction Protocols**
  Although the W3C provides Web-based standards for retrieving and querying machine readable data, they should be extended to cater to usage constraints, such as access policies, intellectual property rights, and privacy preferences. From a service-provision perspective, there is a need to develop policy-aware querying and data retrieval protocols, whereas from an agent interaction perspective, norms should be considered, both in terms of the agent platform and the environment. This raises a number of questions, including:

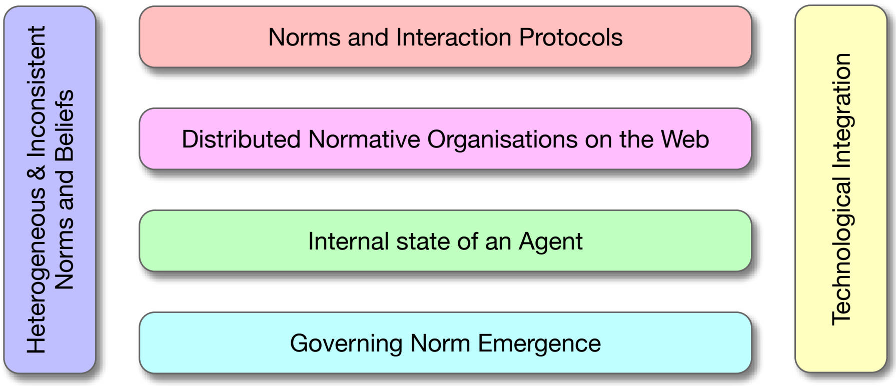


Figure 7 Overview: research challenges for normative MAS and the Web

(i) How can we design norm-aware dynamic interaction protocols?
(ii) Can existing querying and data retrieval protocols be extended, such that they are policy aware?
(iii) What new protocols are needed to facilitate norm governance?

Existing work [32] largely focuses on request-response interactions and imposes restrictions on computation for scenarios involving the interaction of three or more parties [29, 15]. In particular, traditional approaches entwine the control flow details into the protocol, thereby making it difficult to tease apart the content for which a declarative meaning can be specified. Prior work on specifying protocols in terms of norms (commitments) [12, 54] was stymied by the lack of declarative specification of constraints on messages. Recent approaches describe causality and integrity constraints on messages declaratively [64].

- **Distributed normative and open organisations on the web**
  Organisations, institutions or contracts are useful abstractions to structure norms and make them accessible to agents. Although agents have the choice of joining such structures, they may be controlled and be subject to conditions that regulate their admission (and exit), as well as there being an expectation to comply with the organisation's norms. Due to the scale of the Web, several of such permanent, ephemeral or evolving structures may exist, which raises several challenges. First, an agent needs to be able to discover and reason about such organisations and the corresponding norms. The use of ontologies has already been used to facilitate the discovery of services [61] and thus their use in representing organisations is promising. A second challenge is the distributed management of such structures [7], *i.e.*, to monitor and enforce norm compliance. Finally, since the Web is an evolving sociotechnical system, agents require proper abstractions and mechanisms to build and adapt organisations [14].
- **Internal state of an agent and norms**
  Several languages express and support automated reasoning about agent internals, such as beliefs, desires, and intentions. However, challenges exist when it comes to reasoning not only with respect to goals, but also privacy preferences, regulatory constraints, and norms. From a norms perspective, there are several open questions in terms of both specification and enforcement, including:

(i) How can we ensure consistent representation of, and adherence to norms?
(ii) How should a governance architecture be designed in which "rational" agents are incentivised to comply with norms?

(iii) Can a norm violation be excused, based on explanation or beliefs?
(iv) Could transparency facilitate the persuasiveness of the explanation or argument?
(v) How do we ensure an agent is aware of the implications of violating a norm?
(vi) How do we cater for non "rational" (e.g. poorly designed) agents?

- **Governing norm emergence**
  Approaches to the governance of norm emergence are dependent on the capabilities of the agents in a MAS, bearing in mind that population properties may not be homogeneous. We identify three means to govern norm emergence [57] which we classify by adapting the oversight terminology put forward by the [37, §B.II.1.1]:

(i) an external agency observes the behaviour of the population to identify patterns of behaviour and revise the norms to optimise for the system goals (external agent on-the-loop)
(ii) agents propose norm revisions to an external agency, which then implements them subject to an assessment of how those revisions contribute towards system goals (external agent in-command)
(iii) agents propose norm revisions and use an internal decision-making mechanism to establish which changes will be implemented (internal agents in-the-loop).

  In an alternative, decentralized, approach, the norms emerge through the interactions of the agents without an external agency being involved [1]. The question of emergence stems from its role in the norm life cycle. Norm emergence is widely accepted to have happened when a predetermined percentage of the population observes the norm or chooses the same action, and most experimental studies put that percentage at approximately 90%.
- **Heterogeneous and inconsistent norms and beliefs**
  In heterogeneous information system landscapes, one cannot assume that norms and policies are managed and implemented based on a single global specification synchronised across different subsystems, as agents may hold inconsistent beliefs about the norms and policies that apply in any given context. Thus, reaching (partial) agreements in the face of conflicting beliefs regarding norms and policies is an important challenge that needs to be tackled to enable normative distributed MAS on the Web, using, for example, long-running lines of research on agreement technologies [59] and formal argumentation approaches [4].
- **Technological integration**
  To facilitate the practical applicability of research on norms and policies for autonomous agents on the web, it is crucial to build bridges between the technology ecosystems of the different communities. In the Web of Things community, engineering-oriented work is conducted in a highly practice-oriented manner, in close alignment with industry practitioners as well as standardisation bodies such as the W3C. An example of a practice-oriented work is the line of W3C IoT standards that features an abstract architecture description [51] and an interface specification (W3C WoT Scripting API) [49], supported by a JavaScript reference implementation[34].
  Research on engineering autonomous agents and MAS has primarily gained traction within the academic community [56], and standardisation attempts such as FIPA[35] have lacked significant adoption. Adjusting agent-oriented programming and software engineering approaches to better serve the Semantic Web & Linked Data and Web Architecture &

[34] `https://github.com/eclipse/thingweb.node-wot`
[35] `http://www.fipa.org/`.

WoT communities are an opportunity for the MAS community to move their engineering research closer to practice. This can be achieved by:

(i) building general-purpose interfaces to powerful multi-agent programming frameworks like JaCaMo [7] (starting points are, e.g., hypermedia-mas [17], *jacamo-rest* [2]), and/or
(ii) implementing minimally viable agent-oriented abstractions for mainstream programming language ecosystems (e.g., JS-son [45]).

6.4.5 Conclusion

In this chapter we discussed the relevance of norms and policies for governing complex sociotechnical multi-agent systems on the Web. The key challenge – the conceptual and technological integration of normative concepts with WoT abstractions and systematic evaluation of the practical usefulness of the integration results – is aligned with the general challenge for *autonomous agents on the Web* to transfer the rich theoretical achievements of the broader MAS community to the practical and engineering-oriented WoT community, and to facilitate real-world applications at scale.

References

1 N. Ajmeri, H. Guo, P. K. Murukannaiah, and M. P. Singh. Robust norm emergence by revealing and reasoning about context: Socially intelligent agents for enhancing privacy. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 28–34, Stockholm, July 2018. IJCAI.

2 C. J. Amaral, J. F. Hübner, and T. Kampik. Towards Jacamo-rest: A Resource-Oriented Abstraction for Managing Multi-Agent Systems. In G. Vaz Alves, G. Guiménez Lug, A. Pinz Borgeso, and C. E. Pantoja, editors, *Proceedings of the 14th Workshop-School on Agents, Environments, and Applications*, pages 140–151, 2020.

3 G. Andrighetto, G. Governatori, P. Noriega, and L. van der Torre. Normative multi-agent systems (Dagstuhl Seminar 12111). *Dagstuhl Reports*, 2(3), 2012.

4 P. Baroni, D. M. Gabbay, M. Giacomin, and L. van der Torre. *Handbook of formal argumentation*. College Publications, 2018.

5 T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific American*, 2001.

6 G. Boella, L. Van Der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2-3):71–79, 2006.

7 O. Boissier, R. H. Bordini, J. Hübner, and A. Ricci. *Multi-agent oriented programming: programming multi-agent systems using JaCaMo*. MIT Press, 2020.

8 P. A. Bonatti and D. Olmedilla. Rule-based policy representation and reasoning for the semantic web. In *Proceedings of the Third International Summer School Conference on Reasoning Web*. Springer-Verlag, 2007.

9 J. M. Bradshaw, editor. *Software agents*. MIT press, 1997.

10 T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008. W3C Recommendation, World Wide Web Consortium (W3C), Nov. 26 2008. URL `https://www.w3.org/TR/2008/REC-xml-20081126/`.

11 E. Cabrio, A. Palmero Aprosio, and S. Villata. These Are Your Rights. In *The Semantic Web: Trends and Challenges*. Springer International Publishing, 2014.

12 A. Chopra and M. P. Singh. Nonmonotonic commitment machines. In F. Dignum, editor, *Advances in Agent Communication: Proceedings of the International Workshop on Agent Communication Languages (ACL 2003)*, number 2922 in Lecture Notes in Artificial Intelligence, pages 183–200, Melbourne, 2004. Springer.

13 A. Chopra, L. van der Torre, H. Verhagen, and S. Villata. *Handbook of Normative Multiagent Systems*. College Publications, 2018. ISBN 9781848902855.

14 A. K. Chopra and M. P. Singh. From social machines to social protocols: Software engineering foundations for sociotechnical systems. In *Proceedings of the 25th International World Wide Web Conference*, pages 903–914, Montréal, Apr. 2016. ACM.

15 A. K. Chopra, S. H. Christie V, and M. P. Singh. An evaluation of communication protocol languages for engineering multiagent systems. *Journal of Artificial Intelligence Research (JAIR)*, 69:1351–1393, Dec. 2020.

16 A. Cimmino, M. McCool, F. Tavakolizadeh, and K. Toumura. Web of Things (WoT) Discovery, W3C First Public Working Draft 24 November 2020. W3C First Public Working Draft, World Wide Web Consortium (W3C), Nov. 24 2020. URL `http://www.w3.org/TR/2020/WD-wot-discovery-20201124/`.

17 A. Ciortea, O. Boissier, and A. Ricci. Engineering world-wide multi-agent systems with hypermedia. In D. Weyns, V. Mascardi, and A. Ricci, editors, *Engineering Multi-Agent Systems – 6th International Workshop, EMAS 2018, Stockholm, Sweden, July 14-15, 2018, Revised Selected Papers*, volume 11375 of *Lecture Notes in Computer Science*, pages 285–301. Springer, 2018. doi: 10.1007/978-3-030-25693-7_15. URL `https://doi.org/10.1007/978-3-030-25693-7_15`.

18 R. Cyganiak, D. Wood, and M. Lanthaler. RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014. W3C Recommendation, World Wide Web Consortium (W3C), Feb. 25 2014. URL `http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`.

19 DAML-S Coalition:, A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara. DAML-S: Web Service Description for the Semantic Web. In *First International Semantic Web Conference (ISWC) Proceedings*, pages 348–363, 2002.

20 M. Dastani, J. Dix, H. Verhagen, and S. Villata. Normative multi-agent systems (dagstuhl seminar 18171). *Dagstuhl Reports*, 8(4):72–103, 2018.

21 M. De Vos, S. Kirrane, J. Padget, and K. Satoh. ODRL policy modelling and compliance checking. In P. Fodor, M. Montali, D. Calvanese, and D. Roman, editors, *Rules and Reasoning*, pages 36–51, Cham, 2019. Springer International Publishing. ISBN 978-3-030-31095-0.

22 N. Desai, A. U. Mallya, A. K. Chopra, and M. P. Singh. OWL-P: A methodology for business process development. In *Agent-Oriented Information Systems III, 7th International Bi-Conference Workshop, AOIS2005, Utrecht, Netherlands, July 26, 2005, and Klagenfurt, Austria, October 27, 2005, Revised Selected Papers*, number 3529 in Lecture Notes in Computer Science, pages 79–94, Berlin, 2006. Springer.

23 M. d'Inverno, M. Luck, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. Communicating open systems. *Artif. Intell.*, 186:38–94, 2012. doi: 10.1016/j.artint.2012.03.004.

24 P. Doran, V. A. M. Tamma, T. R. Payne, and I. Palmisano. Dynamic selection of ontological alignments: A space reduction mechanism. In C. Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 2028–2033, 2009.

25 M. Esteva, J. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J.-J. C. Meyer and M. Tambe, editors, *Intelligent Agents VIII*, pages 348–366, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45448-9.

26 J. Euzenat and P. Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013. ISBN 978-3-642-38720-3.

27 C. Feier, A. Polleres, R. Dumitru, J. Domingue, M. Stollberg, and D. Fensel. Towards intelligent web services: the web service modeling ontology (WSMO). In *2005 International Conference on Intelligent Computing (ICIC'05)*, 2005.

28 D. Fensel and E. Motta. Structured development of problem solving methods. *IEEE Trans. Knowl. Data Eng.*, 13(6):913–932, 2001. doi: 10.1109/69.971187.

29 A. Ferrando, M. Winikoff, S. Cranefield, F. Dignum, and V. Mascardi. On the enactability of agent interaction protocols: Toward a unified approach. *CoRR*, abs/1902.01131v4:1–13, Feb. 2019.

30 R. T. Fielding. *Representational State Transfer (REST)*. PhD thesis, University of California, Irvine, 2000. Chapter 5 of Roy Fielding's thesis.

31 R. T. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Technical report, IETF, June 2014. URL `http://tools.ietf.org/html/rfc7231`.

32 FIPA. FIPA interaction protocol specifications, 2003. URL `http://www.fipa.org/repository/ips.html`. FIPA: The Foundation for Intelligent Physical Agents.

33 G. Governatori, L. Ho-Pun, A. Rotolo, S. Villata, and F. Gandon. Heuristics for Licenses Composition. *Frontiers in Artificial Intelligence and Applications*, 2013a.

34 G. Governatori, A. Rotolo, S. Villata, and F. Gandon. One license to compose them all. In *International Semantic Web Conference*, 2013b.

35 G. Governatori, H.-P. Lam, A. Rotolo, S. Villata, G. A. Atemezing, and F. L. Gandon. Live: a tool for checking licenses compatibility between vocabularies and data. In *International Semantic Web Conference*, 2014.

36 B. C. Grau and B. Motik. Reasoning over ontologies with hidden content: The import-by-query approach. *J. Artif. Int. Res.*, 45(1):197–255, Sept. 2012. ISSN 1076-9757.

37 High-Level Expert Group on Artificial Intelligence (AI HLEG). Ethics Guidelines for Trustworthy AI, 2019. URL `https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai`.

38 I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. Swrl: A semantic web rule language combining owl and ruleml. W3c member submission, World Wide Web Consortium, 2004. URL `http://www.w3.org/Submission/SWRL`.

39 M. N. Huhns and M. P. Singh. Ontologies for agents. *IEEE Internet Computing (IC)*, 1(6): 81–83, Nov. 1997. Instance of the column *Agents on the Web*.

40 I. Jacobs and N. Walsh. Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004. W3C Recommendation, World Wide Web Consortium (W3C), Dec. 15 2004. URL `http://www.w3.org/TR/2004/REC-webarch-20041215/`.

41 E. Jimenez-Ruiz, T. R. Payne, A. Solimando, and V. Tamma. Limiting logical violations in ontology alignment through negotiation. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. AAAI Press, April 2016.

42 S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay, and M. Kovatsch. Web of Things (WoT) Thing Description, W3C Recommendation 9 April 2020. W3C Recommendation, World Wide Web Consortium (W3C), Apr. 09 2020. URL `https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/`.

43 O. Kafali, N. Ajmeri, and M. P. Singh. DESEN: Specification of sociotechnical systems via patterns of regulation and control. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 29(1), Feb. 2020. ISSN 1049-331X.

44 L. Kagal and T. Finin. A policy language for a pervasive computing environment. In *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, 2003.

45 T. Kampik and J. C. Nieves. Js-son – a lean, extensible javascript agent programming library. In L. A. Dennis, R. H. Bordini, and Y. Lespérance, editors, *Engineering Multi-Agent Systems*, pages 215–234, Cham, 2020. Springer International Publishing. ISBN 978-3-030-51417-4.

46 M. Kifer and H. Boley. Rif overview (second edition). W3c working group note 5 february 2013, World Wide Web Consortium, 2013. URL `https://www.w3.org/TR/rif-overview/`.

47 M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, July 1995. ISSN 0004-5411.

48 S. Kirrane, A. Mileo, and S. Decker. Access control and the resource description framework: A survey. *Semantic Web*, 2017.

49 Z. Kis, D. Peinter, C. Aguzzi, J. Hund, and K. Nimura. Web of Things (WoT) Scripting API, W3C Working Group Note 24 November 2020. W3C Working Group Note, World Wide Web Consortium (W3C), Nov. 24 2020. URL `https://www.w3.org/TR/2020/NOTE-wot-scripting-api-20201124/`.

50 M. Koubarakis. Multi-agent systems and peer-to-peer computing: Methods, systems, and challenges. In M. Klusch, A. Omicini, S. Ossowski, and H. Laamanen, editors, *Cooperative Information Agents VII*, pages 46–61, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45217-1.

51 M. Kovatsch, R. Matsukura, M. Lagally, T. Kawaguchi, K. Toumura, and K. Kajimoto. Web of Things (WoT) Architecture, W3C Recommendation 9 April 2020. W3C Recommendation, World Wide Web Consortium (W3C), Apr. 9 2020. URL `https://www.w3.org/TR/2020/REC-wot-architecture-20200409/`.

52 J. Lee, J. Padget, B. Logan, D. Dybalova, and N. Alechina. N-Jason: Run-time norm compliance in AgentSpeak(L). In F. Dalpiaz, J. Dix, and M. B. van Riemsdijk, editors, *Engineering Multi-Agent Systems – Second International Workshop, EMAS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers*, volume 8758 of *Lecture Notes in Computer Science*, pages 367–387. Springer, 2014. ISBN 978-3-319-14483-2.

53 S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, page 59–66, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918770.

54 A. U. Mallya and M. P. Singh. An algebra for commitment protocols. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 14(2):143–163, Apr. 2007.

55 D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. Sycara. Bringing semantics to web services: The owl-s approach. In J. Cardoso and A. Sheth, editors, *Semantic Web Services and Web Process Composition*, pages 26–42, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-30581-1.

56 V. Mascardi, D. Weyns, A. Ricci, C. B. Earle, A. Casals, M. Challenger, A. Chopra, A. Ciortea, L. A. Dennis, A. F. Díaz, A. El Fallah-Seghrouchni, A. Ferrando, L.-r. Fredlund, E. Giunchiglia, Z. Guessoum, A. Günay, K. Hindriks, C. A. Iglesias, B. Logan, T. Kampik, G. Kardas, V. J. Koeman, J. B. Larsen, S. Mayer, T. Méndez, J. C. Nieves, V. Seidita, B. T. Teze, L. Z. Varga, and M. Winikoff. Engineering multi-agent systems: State of affairs and the road ahead. *SIGSOFT Softw. Eng. Notes*, 44(1):18–28, Mar. 2019. ISSN 0163-5948.

57 A. Morris-Martin, M. De Vos, and J. Padget. Norm emergence in multiagent systems: A viewpoint paper. *Autonomous Agents and Multi-Agent Systems*, 33(6):706–749, 2019.

58 P. Noriega. *Agent Mediated Auctions: The Fishmarket Metaphor*. PhD thesis, Universitat Autònoma de Barcelona, 1997. URL `https://www.iiia.csic.es/research/thesis-details?pastphd_id=85`.

59 S. Ossowski. *Agreement Technologies*. Springer Publishing Company, Incorporated, 2012. ISBN 9400755821.

60 M. Palmirani, G. Governatori, A. Rotolo, S. Tabet, H. Boley, and A. Paschke. Legalruleml: Xml-based rules and norms. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 298–312. Springer, 2011.

61 M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Semantic matching of web services capabilities. In I. Horrocks and J. Hendler, editors, *The Semantic Web – ISWC 2002*, pages 333–347, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-48005-1.

62 E. Reshetova and M. McCool. Web of Things (WoT) Security and Privacy Guidelines, W3C Working Group Note 6 November 2019. W3C Working Group Note, World Wide Web Consortium (W3C), Nov. 6 2019. URL `https://www.w3.org/TR/2019/NOTE-wot-security-20191106/`.

63 M. Riemsdijk, L. Dennis, M. Fisher, and K. Hindriks. Agent reasoning for norm compliance: A semantic approach. In *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, volume 1, pages 499–506, 05 2013.

64 M. P. Singh. Information-driven interaction-oriented programming: BSPL, the Blindingly Simple Protocol Language. In *Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 491–498, Taipei, May 2011. IFAAMAS.

65 M. P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1), Dec. 2013. ISSN 2157-6904.

66 M. P. Singh, A. K. Chopra, and N. Desai. Commitment-based service-oriented architecture. *Computer*, 42(11):72–79, 2009. doi: 10.1109/MC.2009.347.

67 S. Staab and H. Stuckenschmidt, editors. *Semantic Web and Peer-to-Peer – Decentralized Management and Exchange of Knowledge and Information.* Springer, 2006. ISBN 978-3-540-28346-1.

68 P. R. Telang, M. P. Singh, and N. Yorke-Smith. A coupled operational semantics for goals and commitments. *Journal of Artificial Intelligence Research (JAIR)*, 65:31–85, May 2019.

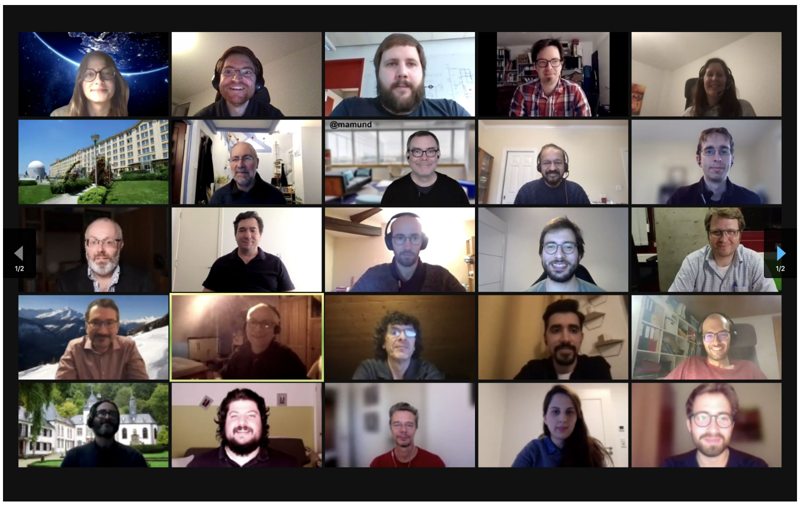
69 Y. B. Udupi and M. P. Singh. Contract enactment in virtual organizations: A commitment-based approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 722–727, Boston, July 2006. AAAI Press.

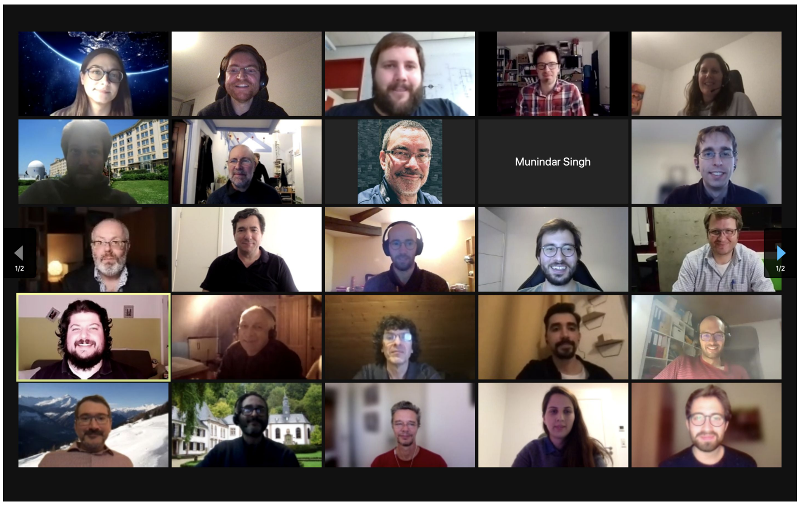
70 C. D. Vescovo, M. Horridge, B. Parsia, U. Sattler, T. Schneider, and H. Zhao. Modular structures and atomic decomposition in ontologies. *J. Artif. Intell. Res.*, 69:963–1021, 2020.

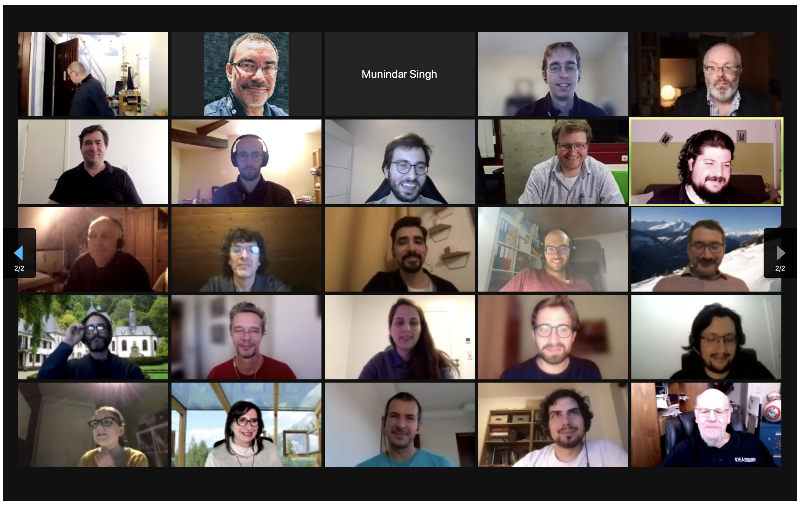
71 S. Villata and F. Gandon. Licenses compatibility and composition in the web of data. In *Third International Workshop on Consuming Linked Data (COLD2012)*, 2012.

Participants

- Cleber Jorge Amaral
Universidade Federal de Santa Catarina, BR
- Mike Amundsen
MuleSoft LLC –
San Francisco, US
- Olivier Boissier
Ecole des Mines – St. Etienne, FR
- Rafael H. Bordini
PUCRS – Porto Alegre, BR
- Arne Bröring
Siemens AG – München, DE
- Samuele Burattini
University of Bologna, IT
- Jean-Paul Calbimonte
HES-SO Valais – Sierre, CH
- Victor Charpenay
Mines Saint-Étienne, FR
- Samuel Christie
Lancaster University, GB
- Andrei Ciortea
Universität St. Gallen, CH
- Olivier Corby
INRIA – Sophia Antipolis, FR
- Angelo Croatti
Università di Bologna, IT
- Jérôme Euzenat
INRIA – Grenoble, FR
- Matthias Farnbauer-Schmidt
Universität Erlangen-Nürnberg, DE
- Fabien Gandon
INRIA – Sophia Antipolis, FR
- Dominique Guinard
EVRYTNG – London, GB
- Andreas Harth
Fraunhofer IIS – Nürnberg, DE
- James A. Hendler
Rensselaer Polytechnic Institute – Troy, US
- Jomi Fred Hübner
Federal University of Santa Catarina, BR
- Tobias Käfer
KIT – Karlsruher Institut für Technologie, DE
- Lalana Kagal
MIT – Cambridge, US
- Timotheus Kampik
University of Umeå, SE
- Sabrina Kirrane
Wirtschaftsuniversität Wien, AT
- Ege Korkan
TU München, DE
- Matthias Kovatsch
Huawei Technologies –
München, DE
- Maxime Lefrançois
Mines Saint-Étienne, FR
- Adnane Mansour
Mines Saint-Étienne, FR
- Simon Mayer
Universität St. Gallen, CH
- Mahda Noura
TU Chemnitz, DE
- Julian Padget
University of Bath, GB
- Cesare Pautasso
University of Lugano, CH
- Terry Payne
University of Liverpool, GB
- María Poveda-Villalón
Technical University of Madrid, ES
- Dave Raggett
W3C – United Kingdom, GB
- Alessandro Ricci
Università di Bologna, IT
- Sebastian Schmid
Universität Erlangen-Nürnberg, DE
- Daniel Schraudner
Universität Erlangen-Nürnberg, DE
- Ronny Seiger
Universität St. Gallen, CH
- Munindar P. Singh
North Carolina State University – Raleigh, US
- Ruben Taelman
Ghent University, BE
- Valentina Tamma
University of Liverpool, GB
- Danai Vachtsevanou
Universität St. Gallen, CH
- Ruben Verborgh
Ghent University, BE
- Erik Wilde
Axway – Zürich, CH
- Antoine Zimmermann
Ecole des Mines –
St. Etienne, FR




Munindar Singh


Munindar Singh