Harnessing Hypermedia MAS and Microservices to Deliver Web Scale Agent-based Simulations

Rem Collier[a], Seán Russell[b] and Fatemeh Golpayegani[c]
School of Computer Science, University College Dublin, Ireland

Keywords: Semantic Web, Hypermedia Systems, Agent-based Modelling, Microservices.

Abstract: This paper presents a vision for a new breed of Agent-Based Simulations that are built on the technology of the Web. Inspired by the emergence of the recently proposed Hypermedia Multi-Agent Systems concept - which combines the concepts Hypermedia Systems, Semantic Web and Affordances - we propose a novel approach to implementing complex agent-based simulations built from suites of loosely-coupled reusable components in a manner that ensures scalability.

1 INTRODUCTION

Agent-Based Modelling (ABM) is a bottom-up approach to studying the behaviour of complex systems (Polhill et al., 2019) that has been successfully applied in many domains, including: Logistics (Du et al., 2019), Intelligent Transportation Systems (Golpayegani et al., 2018) and Power Systems (Teixeira et al., 2020). Such systems are modelled as collections of agents; each one encapsulating private state and behaviour. Global system behaviours emerge through interactions between the constituent agents.

A key challenge in this area is the ability to model systems of increasing complexity. It is acknowledged that such systems are becoming too complex to be captured in a single model (Kitova et al., 2016). One solution is to use Hybrid Simulation (HS) (Eldabi et al., 2018). Broadly, HS combine multiple interconnected sub-simulations, potentially implemented using a diverse set of modelling techniques (Mustafee et al., 2017). It is increasingly used in Operational Research (Brailsford et al., 2019) and Socio-Environmental Systems (Turner II et al., 2016).

The development of HS tools and methodologies is still an open research problem. In (Polhill et al., 2019), the authors highlight the lack of suitable tools and frameworks for integrating ABM with other techniques. Interoperability is a particular issue leading to many existing HS being built using a single tool (Eldabi et al., 2018).

This paper argues that HS should not be built on monolithic architectures and homogeneous technology stacks, but should instead be implemented as loosely-coupled collections of reusable components, written using diverse programming languages and frameworks, that are designed to be deployed at scale. In essence, it argues that HS should be implemented using a microservices architecture (Fowler, 2014).

Section 3 presents a vision of a new framework for implementing HS based on ABM, where each sub-simulation is encapsulated as a microservice that uses REpresentational State Transfer (REST) to serve simulation semantically enriched state representations. The agent layer, which consumes this state, is implemented as a *Hypermedia MAS* (Ciortea et al., 2019b). Section 4 presents some challenges and opportunities for further research, and section 5 presents some concluding remarks.

2 THE CURRENT STATE OF PLAY

There are a range of approaches to implementing ABMs, and (Abar et al., 2017) provides an excellent review of them. What is clear from the review is that ABM has traditionally been viewed as a desktop computer style of exercise where tools are provided to support the creation and execution of models on a single machine. Such tools typically consist of some mechanism to specify agent types and be-

[a] https://orcid.org/0000-0003-0319-0797
[b] https://orcid.org/0000-0003-1992-8303
[c] https://orcid.org/0000-0002-3712-6550

Collier, R., Russell, S. and Golpayegani, F.
Harnessing Hypermedia MAS and Microservices to Deliver Web Scale Agent-based Simulations.
DOI: 10.5220/0010711100003058
In *Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST 2021)*, pages 404-411
ISBN: 978-989-758-536-4; ISSN: 2184-3252

haviours, the environment(s) that the agent inhabits, any constraints on interaction, and any rules for adaptation of behaviour. The main constraint on the use of such tools to create simulations is often the time taken for the simulation to be executed, which can lead to repeated simplifications of the model until the execution time is acceptable (Taylor et al., 2020).

2.1 Simulating at Scale

Within the simulation community, two main approaches have emerged to overcoming the limits of desktop simulation. The Distributed Simulation (DS) community (Rashid et al., 2018) focuses on the development of techniques that can be deployed on high-performance computing clusters. This has led to the development of tools, such as: RePAST HPC (Collier and North, 2012) and MATSim (W Axhausen et al., 2016). Taylor (Taylor, 2019) reviews DS through the lens of Operational Research, highlighting the prevalence of bespoke implementations tailored to specific scenarios, the lack of model reuse and the need for well-designed frameworks.

A criticism of DS is the cost and availability of computing clusters. This led to the emergence of Cloud Based Simulation (CBS) (Taylor, 2019) which is concerned with the deployment of simulations in the cloud. Example ABM tools include: CloudSME (Taylor et al., 2018), RISE (Al-Zoubi and Wainer, 2013) and Distributed Mason (Cordasco et al., 2018), a port of Mason (Luke et al., 2004) for the cloud.

In (Hüning et al., 2016), the authors describe MARS, a cloud-based ABM tool that uses a layered model adapted from Geographical Information Systems, where each layer represents a distinct feature of the simulation. It supports the distribution of models; each layer is deployed as a separate node and individual layers can be distributed across multiple nodes.

More recently, the CBS community has started exploring the intersection of microservices and simulation. Microservices are an architectural style that promote the decomposition of complex systems into distributed applications composed of simple services that are designed to scale (Fowler, 2014; Zimmermann, 2017). In (Taylor et al., 2020), the authors introduce a microservices framework for the creation of deadline based simulations. Their prototype implemented an auto-scaling mechanism that runs multiple REPAST simulations in parallel to meet deadlines (Anagnostou et al., 2019). This is achieved by modelling each simulation as a microservice.

Finally, (Pump et al., 2019) describes the only known project that has used microservices to create a HS from heterogeneous models. However, as the authors state, the developed model was bespoke and tailored to the specific problem they were addressing.

The adoption of microservices in the design of simulators is appealing for a number of reasons:

- they promote the development of small, loosely-coupled systems that maintain their own independent state (Dragoni et al., 2018);
- they are deployed within an ecosystem of tools and components that facilitate rapid and agile development techniques, are easy to extend and support automated management of fault tolerance and scaling (Richards, 2015); and
- they support the integration of heterogeneous systems built using diverse technology stacks (Thönes, 2015).

This is especially true of Hybrid Simulations, where bounded context, isolation and loose coupling (Zimmermann, 2017) promotes the decomposition of simulators into discrete components that can be developed and deployed independently.

The decomposition of simulators into components offers the potential for their reuse across multiple scenarios. Microservices facilitate this through the requirement that all the components adhere to a uniform interface, engendering the use of diverse technology stacks. This presents a route towards the development of hybrid simulations that combine not only different programming languages and tool sets, but also integrate different simulation techniques.

2.2 Simulating Intelligence

A recent survey on the use of the Belief-Desire-Intention (BDI) architecture (Rao et al., 1995) in social simulation (Adam and Gaudou, 2016) highlights the potential quality improvements that cognitive architectures bring to simulation. They outline an emerging trend in social simulation that argues for the application of the "Keep It Descriptive, Stupid" (KIDS) principle over the "Keep It Simple, Stupid" (KISS) principle. Their argument is that, with increased computing resources there is less need to use simplistic agent models in an effort to maximise the performance of the simulation. A better solution is to use richer cognitive agent architectures to facilitate the creation of more nuanced models.

Agent-Oriented Programming (AOP) languages (Shoham, 1993) offer such rich architectures through logic based programming constructs that are both verifiable and explainable (Kravari and Bassiliades, 2015). Recently, there has been a resurgence of interest in using AOP for simulation (Bădică et al., 2018b; Lawlor et al., 2018; Bădică et al., 2018a; Balabanov et al., 2020). Other work has explored how to support discrete event simulation using AOP. In (Larsen, 2019a) the authors describe a simulator tool based on GAMA (Taillandier et al., 2016) that is applied to Hospital staff planning (Larsen, 2019b). In (Ricci et al., 2020) an emerging simulation platform based on JaCaMo (Boissier et al., 2013) is described. Finally, in (Muto et al., 2020) a Socio-Ecological Systems approach to modelling the agricultural economy of the Rancherina River Basin using the BESA (González et al., 2003) agent toolkit is presented.

2.3 Affordances in Simulation

Recent research suggests that affordances may provide a more flexible approach to ABM (Klügl, 2015). The concept of an affordance originates in ecological psychology as a means of representing the relationship between environmental objects and the potential actions that an agent (human or otherwise) may perform with those objects (Gibson, 1979).

Affordances are information perceived from the environment that signifies that a particular action may be performed. They allow for a higher level of abstraction in agent-environment interactions, allowing an agent to reason about the actions it can perform instead of having hard coded actions in plans.

The use of affordances in ABM parallels the changing perception of agent-environment interaction where the environment now viewed as an explicit part of the MAS (Weyns et al., 2007). This view is being realised through systems such as EIS (Behrens et al., 2012) and CArtAgO (Ricci et al., 2007), which provide an abstraction of the environment that can be used across agent platforms. CArtAgO has already been used in simulations systems, JaCaMo-sim platform (Ricci et al., 2020) is based on its use, and also in combination with affordances in a Web of Things environment (Ciortea et al., 2018b).

Affordances-effect pairs have been utilised in modelling of human behaviour in complex environments (Jooa et al., 2013). Affordance fields have been used to represent the suitability of potential actions available to an agent at a given time in path planning simulations (Kapadia et al., 2009). Affordances have also been used in traffic simulation (Ksontini et al., 2015) and simulating the behaviour of tanks in a capture the flag exercise (Papasimeon, 2010).

Each of these examples represents a bespoke implementation of the concept of affordances in ABM, each choosing how to represent affordances, how to fit them into the execution of the agent, and how to represent them in the language. The benefits of using affordances in ABM can be greatly enhanced, or the cost of implementation diminished, by the development of a standard representation. This would enable greater interoperability between agents and simulation systems, between agents and environments, and between simulation systems.

3 A NEW VISION FOR SIMULATION AT SCALE

We believe there is a need for Hybrid Simulations that can scale while being built from reusable components that are designed for interoperability. The combination of techniques introduced in this paper represent a potential pathway to achieving this. Our approach is inspired by the successes achieved by leading technology companies, such as Netflix and Amazon, who have met the challenge of deploying their infrastructures at Web Scale.

Underpinning their success is their adoption of the microservices architecture (Fowler, 2014). As is highlighted in section 2.1, little research has been carried out into the use of microservices in simulation. While the approach we propose is focused on the application of microservices to ABM, we believe that it can be used to support the creation of simulations that combine diverse modelling approaches.

3.1 Microservices and Simulation

In our view, microservices are ideally suited to the implementation of Hybrid Simulations (HS). The mapping of microservices to sub-simulations is in keeping with the ethos of the approach. In fact, as is discussed in section 2.1, adopting a microservices perspective brings a range of additional benefits including mature tool ecosystems and polyglot computing.

The notion of polyglot computing sits well with HS as it is expected that such systems will be composed of multiple sub-simulations implemented using heterogeneous modelling techniques and languages. However, it does not address how to engender interoperability between those sub-simulations. It is our view that this can be achieved through the adaptation of the way that Linked Data is currently used in the Web of Things (WoT) (Guinard and Trifa, 2016). Broadly speaking, Linked Data is an approach to realising Tim Berners Lee's vision of the Semantic Web (Berners-Lee et al., 2001) through the creation of typed links. These links are embedded within documents that represent data from different Web sources. Critically, Linked Data supports machine readability through the use of ontologies, encoded and interpreted using Semantic Web technologies (Bizer et al., 2011).

Within the WoT, Linked Data has been used to help develop a set of standards, known as *Thing Descriptions* (Charpenay and Käbisch, 2020). These are machine readable documents that describe the ”things” that have been deployed. They describe the capabilities of the devices, allowing clients to reason about how (or whether) to use them. The availability of such descriptions is driving research into a range of ”thing” composition techniques, including: thing discovery (Zhou et al., 2016), Web Mashups (Guinard and Trifa, 2009), automated composition tools (Noura and Gaedke, 2019) and even agent-based approaches (Savaglio et al., 2020). Such descriptions can also be used with manual composition tools.

Our vision is inspired by this view. We believe that a *Simulation Description* that is analogous to Thing Descriptions would promote interoperability and reuse of sub-simulations. Such a description could be used both to specify the nature of the environment as well as the inputs and outputs associated with it. The application of semantic meaning to all data inputs and outputs related to each simulation would allow for their integration into data processing pipelines for tasks such as transforming data between representations or handling (time) scaling issues.

3.2 Microservices and Agent-based Simulation

In our view, there are two main ways in which ABM can be integrated with the Hybrid Simulation framework proposed above. One approach is to view them simply as self-contained sub-simulations that are integrated into larger simulations that are not agent-based. While there are a number of challenges in achieving this, it is not the focus of the vision we present in this paper. In our view, such an approach can be realised through existing simulation frameworks.

Instead, our vision is oriented towards a second approach where ABM plays a key role in the overall simulation. Here, the ABM acts as a central component in which agents are linked with multiple sub-simulations, developed using diverse simulation techniques, whose state is consumed by the agents so that they may more accurately model the behaviour of the population they are simulating. Due to the potential complexity arising from processing multiple state data streams, we believe that the simple agent models traditionally used in ABM will be inadequate and more nuanced models are required. In our view, such an approach is congruent with the ”Keep It Descriptive, Stupid” principle described in (Adam and Gaudou, 2016). The authors use this principle to argue that the social simulation community needs to adopt BDI style models, tools and programming languages.

We agree with this sentiment and argue that it is essential for managing the complexity emerging from agents interacting with multiple sub-simulations as is proposed within our vision. In our view, the best way to deliver this is through the use of *Hypermedia MAS*.

In (Ciortea et al., 2018a), the authors introduce the concept of *Hypermedia MAS* as an approach to building dynamic, open and long-lived MAS (Vachtsevanou et al., 2020) that are designed to inter-operate seamlessly with the World Wide Web. In (Ciortea et al., 2019b), this approach is expanded by outlining a vision in which agents are integrated into the hypermedia fabric of the web, and that by doing so, enter into a shared hypermedia environment that is based on the open standards of the Web. In such an environment, devices and physical services can be exposed as first class entities. Hyperlinks and hypermedia controls can be used to discover and interact with those entities or even other agents regardless of their location. Such controls can be published through hypermedia documents that are adapted based on the state of the underlying entities.

Through these hypermedia documents, agents are able to discover at run-time the capabilities of the entities in their environment. Presented appropriately, such a document could be linked to the concept of affordances as described in section 2.3 enabling them to understand not just what the state of the entity is, but also what they are currently able to do to that entity. It is important to understand that the concept of Thing Descriptions, as discussed in section 3.1 is an example of just such a document.

As described earlier, our vision views ABM as the integration point for multiple sub-simulations. This is achieved by viewing the sub-simulations as being part of the environment. The agents would then interact with the sub-systems through an agent-environment interface. In our approach, the environment would be decomposed into a set of microservices and the agents would interact via the APIs exposed by these microservices. There are two possible integration strategies to achieve this: the use of a single microservice that acts as the intermediary between the agents and the subsystems - in the microservices world, this would be considered an API Gateway (Montesi and Weber, 2016); and the direct integration of the agents with multiple linked microservices.

We are especially interested in the latter approach as we believe that it offers a more decentralised and scalable solution. The view also has many parallels with the recent work done on applying Hypermedia MAS to the Web of Things (Ciortea et al., 2019a). In their approach, the highlight the use of *Thing De-*

scriptions as a means for describing *interaction affordances* that can be used by the agent to understand its options. We aim to mirror this approach through the introduction of an *Entity Description* that describes the interface between each sub-simulation and the environment. Agents would use this to configure themselves as they connect to a sub-simulation. We term the interface defined by the entity description to be an *Affordance API*. Finally, the entity descriptions should be linked to corresponding *Simulation Descriptions* as defined in section 3.1.

Mirroring the decomposition of the environment into a set of microservices, we believe that the agents should also be deployed across a set of microservices. A possible solution for achieving this is *Multi-Agent MicroServices (MAMS)* approach (Collier et al., 2019; O'Neill et al., 2020), which has been demonstrated in the ASTRA agent programming language (Collier et al., 2015; Dhaon and Collier, 2014).

In summary, this section expands our vision for a Hybrid Simulation framework to require the use of Hypermedia MAS to implement the agent component of an ABM. Inspired by recent work in the area of WoT, the agents would interact with sub-simulations, modelled as environment microservices, using REST and Linked Data. As with the WoT, agent-environment interactions are governed through the use of interaction affordances.

3.3 Illustrating the Vision

To illustrate our vision, we present a sketch of a Hybrid Simulation of a shopping centre. The simulation, whose architecture is presented in Figure 1 aims to model the occupancy and transit of people through the shopping centre.

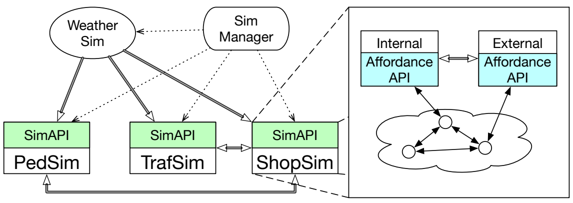

Figure 1: Representation of microservice-based Hypermedia MAS Simulation.

As can be seen, the simulation consists of a number of sub-simulations. The three primary simulations are: **PedSim** which models the arrival and departure of pedestrians on foot; **TrafSim** which does the same for pedestrians who travel by vehicle; and **ShopSim**, which models the shopping center. The idea here, is that **PedSim** and **TrafSim** act as sinks and sources of agents (who model the population) that enter/leave the shopping center (modelled as **ShopSim**). There is no requirement that these first two simulations be agent-based; they could be statistical simulations, that drive the creation of agents inside **ShopSim**, or they could themselves be agent-based simulations. An important challenge to be resolved here, would be how to support both types of simulation without requiring any changes to **ShopSim**. For example, an intermediary component could be added that acts as an interface to **ShopSim**, creating and destroying agents as appropriate. These simulations themselves, could be linked to other simulations, for example, in Figure 1, all three are linked to a weather simulator.

The agent-based **ShopSim** simulator is itself decomposed into two sub-simulations that cover the internal and external locations of the shopping centre respectively. The shopping centre itself can be represented as a graph, with nodes being modelled as endpoint associated with the sub-simulations, and edges modelled as hyperlinks between nodes. Depending on the nature of the environment, these hyperlinks could be either manipulated by the agents themselves, or used internally by the simulations. For example, an agent could transition from a corridor to a shop by submitting a DELETE request to the corridor endpoint and a POST request to the shop endpoint. Alternatively, the agent submit an "enterShop" action to the corridor simulation resulting in the corridor simulator transitioning the agent to the shop. Which approach to use depends largely on the nature of the simulation and whether or not the agent is embodied in the environment. If the agent has a body, then we believe that the latter approach is more appropriate as the representation of the body would need to be transferred from one simulation to the other. This brings more challenges in terms of how to represent the body in a way that does not limit interoperability, and how to standardise transitioning agents between simulations.

Another challenge to be addressed is agent-simulation interaction. Specifically, we are concerned with how the agent is made aware of the actions it can currently perform. As discussed in section 3.2, we believe this is best achieved through the implementation of an *Affordances API*. Both approaches described in the previous paragraph fit this. For simulations where the agent is disembodied, we envisage using equivalents of the WoT Thing Description. Alternatively, in cases where the agent is embodied, the body provides a context for the agent in the simulation. This context can be used to identify the potential actions of the agent, which in turn can be passed to the agent.

Finally, the sim manager is responsible for setting up or managing the overall simulation. Interaction with each sub-simulation is through the SimAPI.

4 CHALLENGES AND OPPORTUNITIES

Realising the vision presented in this paper raises a number of challenges that also present opportunities for the Multi-Agent Systems community. Some of the key challenges are described below:

Tools for Building Hypermedia MAS: Suitable tools for building Hypermedia MAS do not currently exist (Ciortea et al., 2019b). Such tools would need to be hypermedia friendly and provide mechanisms to allow agents to reason with and act on semantic knowledge.

Defining Ontologies for Describing Simulations: The effective description of simulation components is critical to enable their composition and reuse. Development of a standardised suite of ontologies to support this is an essential community activity.

Exploring the Methods and Protocols Needed to Deliver Hybrid Simulation: Before effective standards can emerge, there is a need to explore how to combine simulation components: how component connections should be presented to agents and how an agent should migrate from one component to another. It is concerned with the creation of the methods and protocols needed to allow agents to operate seamlessly across them.

Building Simulations that Can Scale: Being able to decompose a simulation into constituent microservice is not enough. There is a need to understand how to do this in a way that enables scalability. This challenge is further complicated by the need to consider multiple sub-simulations and the need for mechanisms that can knit the individual sub-simulations into a larger whole.

Reusability across Simulation Domains: Understanding how to make sub-simulations reusable across problem domains is another key challenge. This requires the development of standards for defining simulations and techniques for composing them. Some discussion on potential approaches is presented in section 3.1.

5 CONCLUDING REMARKS

This paper presents a vision of a microservices-based approach to implementing Hybrid Simulations that include an ABM component that is underpinned by the recently proposed notion of *Hypermedia MAS*. We have illustrated our vision through a simple example and outlined a number of challenges and opportunities for future research.

REFERENCES

Abar, S., Theodoropoulos, G. K., Lemarinier, P., and O'Hare, G. M. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33.

Adam, C. and Gaudou, B. (2016). Bdi agents in social simulations: a survey. *The Knowledge Engineering Review*, 31(3):207–238.

Al-Zoubi, K. and Wainer, G. (2013). Rise: A general simulation interoperability middleware container. *Journal of Parallel and Distributed Computing*, 73(5).

Anagnostou, A., Taylor, S. J., Abubakar, N. T., Kiss, T., DesLauriers, J., Gesmier, G., Terstyanszky, G., Kacsuk, P., and Kovacs, J. (2019). Towards a deadline-based simulation experimentation framework using micro-services auto-scaling approach. In *2019 Winter Simulation Conference (WSC)*, pages 2749–2758.

Bădică, A., Bădică, C., Buligiu, I., and Ciora, L. (2018a). Devs modeling and simulation using bdi agents: Preliminary considerations. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pages 1–8.

Bădică, A., Bădică, C., Ivanović, M., and Dănciulescu, D. (2018b). Multi-agent modelling and simulation of graph-based predator–prey dynamic systems: A bdi approach. *Expert Systems*, 35(5):e12263.

Balabanov, K., Cejrowski, T., Logofătu, D., and Bădică, C. (2020). Study on population dynamics for triple-linked food chain using a simulation-based approach. *Evolving Systems*, 11(2):215–226.

Behrens, T., Hindriks, K. V., Bordini, R. H., Braubach, L., Dastani, M., Dix, J., Hübner, J., and Pokahr, A. (2012). An Interface for Agent-Environment Interaction. In Collier, R., Dix, J., and Novák, P., editors, *Programming Multi-Agent Systems*, volume 6599 of *Lecture Notes in Computer Science*, pages 139–158. Springer Berlin Heidelberg.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):34–43.

Bizer, C., Heath, T., and Berners-Lee, T. (2011). Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global.

Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.

Brailsford, S. C., Eldabi, T., Kunc, M., Mustafee, N., and Osorio, A. F. (2019). Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research*, 278(3).

Charpenay, V. and Käbisch, S. (2020). On modeling the physical world as a collection of things: The w3c thing description ontology. In *European Semantic Web Conference*, pages 599–615. Springer.

Ciortea, A., Boissier, O., and Ricci, A. (2018a). Engineering world-wide multi-agent systems with hypermedia. In *International Workshop on Engineering Multi-Agent Systems*, pages 285–301. Springer.

Ciortea, A., Mayer, S., Boissier, O., and Gandon, F. (2019a). Exploiting Interaction Affordances: On Engineering Autonomous Systems for the Web of Things. In *Second W3C Workshop on the Web of Things The Open Web to Challenge IoT Fragmentation*, Munich, Germany.

Ciortea, A., Mayer, S., Gandon, F., Boissier, O., Ricci, A., and Zimmermann, A. (2019b). 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*.

Ciortea, A., Mayer, S., and Michahelles, F. (2018b). Repurposing Manufacturing Lines on the Fly with Multi-Agent Systems for the Web of Things. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 813–822, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Collier, N. and North, M. (2012). Repast hpc: A platform for large-scale agent-based modeling. *Large-Scale Computing*, 10:81–109.

Collier, R. W., O'Neill, E., Lillis, D., and O'Hare, G. (2019). Mams: Multi-agent microservices. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 655–662.

Collier, R. W., Russell, S., and Lillis, D. (2015). Reflecting on agent programming with agentspeak (l). In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 351–366. Springer.

Cordasco, G., Scarano, V., and Spagnuolo, C. (2018). Distributed mason: A scalable distributed multi-agent simulation environment. *Simulation Modelling Practice and Theory*, 89:15–34.

Dhaon, A. and Collier, R. W. (2014). Multiple inheritance in agentspeak (l)-style programming languages. In *Proceedings of the 4th International Workshop on Programming based on Actors Agents & Decentralized Control*, pages 109–120.

Dragoni, N., Lanese, I., Larsen, S. T., Mazzara, M., Mustafin, R., and Safina, L. (2018). Microservices: How to make your application scale. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Du, J., Jing, H., Choo, K.-K. R., Sugumaran, V., and Castro-Lacouture, D. (2019). An ontology and multi-agent based decision support framework for prefabricated component supply chain. *Information Systems Frontiers*, pages 1–19.

Eldabi, T., Brailsford, S., Djanatliev, A., Kunc, M., Mustafee, N., and Osorio, A. F. (2018). Hybrid simulation challenges and opportunities: a life-cycle approach. In *2018 Winter Simulation Conference (WSC)*, pages 1500–1514. IEEE.

Fowler, M. (2014). MicroServices: A definition of this new architectural term.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton-Mifflin.

Golpayegani, F. et al. (2018). Co-ride: Collaborative preference-based taxi-sharing and taxi-dispatch. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 864–871. IEEE.

González, E., Avila, J., and Bustacara, C. (2003). Besa: Behavior-oriented, event-driven, social-based agent framework. In *PDPTA*, volume 3, pages 1033–1039.

Guinard, D. and Trifa, V. (2009). Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, volume 15, page 8.

Guinard, D. D. and Trifa, V. M. (2016). *Building the web of things*, volume 3. Manning Publications Shelter Island.

Hüning, C., Adebahr, M., Thiel-Clemen, T., Dalski, J., Lenfers, U., and Grundmann, L. (2016). Modeling & simulation as a service with the massive multi-agent system mars. In *Proceedings of the Agent-Directed Simulation Symposium*, pages 1–8.

Jooa, J., Kimb, N., Wyskc, R. A., Rothrockd, L., Sone, Y.-J., Ohb, Y.-g., and Leef, S. (2013). Agent-based simulation of affordance-based human behaviors in emergency evacuation. *Simulation Modelling Practice and Theory*, 32:99–115.

Kapadia, M., Singh, S., Hewlett, W., and Faloutsos, P. (2009). Egocentric Affordance Fields in Pedestrian Steering. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, pages 215–223, New York, NY, USA. Association for Computing Machinery.

Kitova, O. V., Kolmakov, I. B., Dyakonova, L. P., Grishina, O. A., Danko, T. P., and Sekerin, V. D. (2016). Hybrid intelligent system of forecasting of the socio-economic development of the country. *International Journal of Applied Business and Economic Research*, 14(9):5755–5766.

Klügl, F. (2015). Affordance-Based Interaction Design for Agent-Based Simulation Models. In Bulling, N., editor, *Multi-Agent Systems*, pages 51–66, Cham. Springer International Publishing.

Kravari, K. and Bassiliades, N. (2015). A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11.

Ksontini, F., Mandiau, R., Guessoum, Z., and Espié, S. (2015). Affordance-based agent model for road traffic simulation. *Autonomous Agents and Multi-Agent Systems*, 29(5):821–849.

Larsen, J. B. (2019a). Going beyond bdi for agent-based simulation. *Journal of Information and Telecommunication*, 3(4):446–464.

Larsen, J. B. (2019b). *Hospital Staff Planning with Multi-Agent Goals*. PhD thesis, Department of Applied Mathematics and Computer Science, Technical University of Denmark.

Lawlor, F., Collier, R., and Nallur, V. (2018). Towards a programmable framework for agent game playing. *arXiv preprint arXiv:1807.08545*.

Luke, S., Cioffi-Revilla, C., Panait, L., and Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 swarmfest workshop*, volume 8, pages 316–327. Michigan, USA.

Montesi, F. and Weber, J. (2016). Circuit breakers, discovery, and api gateways in microservices. *arXiv preprint arXiv:1609.05830*.

Mustafee, N., Brailsford, S., Djanatliev, A., Eldabi, T., Kunc, M., and Tolk, A. (2017). Purpose and benefits of hybrid simulation: contributing to the convergence of its definition. In *2017 Winter Simulation Conference (WSC)*, pages 1631–1645. IEEE.

Muto, T. J., Bolivar, E. B., and González, E. (2020). Bdi multi-agent based simulation model for social ecological systems. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 279–288. Springer.

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

O'Neill, E., Lillis, D., O'Hare, G. M., and Collier, R. W. (2020). Explicit modelling of resources for multi-agent microservices using the cartago framework. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*.

Papasimeon, M. (2010). *Modelling agent-environment interaction in multi-agent simulations with affordances*. Phd, Defence Science and Technology Organisation, Air Operations Division.

Polhill, J. G., Ge, J., Hare, M. P., Matthews, K. B., Gimona, A., Salt, D., and Yeluripati, J. (2019). Crossing the chasm: a 'tube-map'for agent-based social simulation of policy scenarios in spatially-distributed systems. *GeoInformatica*, 23(2):169–199.

Pump, R., Koschel, A., and Ahlers, V. (2019). Applying microservices principles to simulation tools. In *Service Computation, 11th International Conference on Advanced Service Computing*.

Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: from theory to practice. In *Icmas*, volume 95.

Rashid, Z. N., Zebari, S. R., Sharif, K. H., and Jacksi, K. (2018). Distributed cloud computing and distributed parallel computing: A review. In *2018 International Conference on Advanced Science and Engineering (ICOASE)*, pages 167–172. IEEE.

Ricci, A., Croatti, A., Bordini, R., Hübner, J., and Boissier, O. (2020). Exploiting simulation for mas programming and engineering-the jacamo-sim platform. In *8th International Workshop on Engineering Multi-Agent Systems (EMAS 2020)*.

Ricci, A., Viroli, M., and Omicini, A. (2007). CArtAgO: A Framework for Prototyping Artifact-Based Environments in MAS. In Weyns, D., Parunak, H., and Michel, F., editors, *Environments for Multi-Agent Systems III*, volume 4389 of *Lecture Notes in Computer Science*, pages 67–86. Springer Berlin Heidelberg.

Richards, M. (2015). *Microservices vs. service-oriented architecture*. O'Reilly Media.

Savaglio, C., Ganzha, M., Paprzycki, M., Bădică, C., Ivanović, M., and Fortino, G. (2020). Agent-based internet of things: State-of-the-art and research challenges. *Future Generation Computer Systems*, 102:1038–1053.

Shoham, Y. (1993). Agent-oriented programming. *Artificial intelligence*, 60(1):51–92.

Taillandier, P., Bourgais, M., Caillou, P., Adam, C., and Gaudou, B. (2016). A bdi agent architecture for the gama modeling and simulation platform. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 3–23. Springer.

Taylor, S., Anagnostou, A., Abubakar, N., Kiss, T., Deslauriers, J., Terstyanszky, G., Kacsuk, P., Kovacs, J., Kite, S., Pattison, G., et al. (2020). Innovations in simulation: Experiences with cloud-based simulation experimentation. In *Winter Simulation Conference 2020*.

Taylor, S. J. (2019). Distributed simulation: state-of-the-art and potential for operational research. *European Journal of Operational Research*, 273(1):1–19.

Taylor, S. J., Kiss, T., Anagnostou, A., Terstyanszky, G., Kacsuk, P., Costes, J., and Fantini, N. (2018). The CloudSME simulation platform and its applications: A generic multi-cloud platform for developing and executing commercial cloud-based simulations. *Future Generation Computer Systems*, 88:524–539.

Teixeira, B., Santos, G., Pinto, T., Vale, Z., and Corchado, J. M. (2020). Application ontology for multi-agent and web-services' co-simulation in power and energy systems. *IEEE Access*, 8:81129–81141.

Thönes, J. (2015). Microservices. *IEEE software*, 32(1):116–116.

Turner II, B., Esler, K. J., Bridgewater, P., Tewksbury, J., Sitas, N., Abrahams, B., Chapin III, F. S., Chowdhury, R. R., Christie, P., Diaz, S., et al. (2016). Socio-environmental systems (ses) research: what have we learned and how can we use this information in future research programs. *Current opinion in environmental sustainability*, 19:160–168.

Vachtsevanou, D., Junker, P., Ciortea, A., Mizutani, I., and Mayer, S. (2020). Long-lived agents on the web: Continuous acquisition of behaviors in hypermedia environments. In *Companion Proceedings of the Web Conference 2020*, pages 185–189.

W Axhausen, K., Horni, A., and Nagel, K. (2016). *The multi-agent transport simulation MATSim*. Ubiquity Press.

Weyns, D., Omicini, A., and Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1).

Zhou, Y., De, S., Wang, W., and Moessner, K. (2016). Search techniques for the web of things: A taxonomy and survey. *Sensors*, 16(5):600.

Zimmermann, O. (2017). Microservices tenets. *Computer Science-Research and Development*, 32(3-4).