

Visualizing Multi-Agent Systems

Peter J. Polack, Jr.

Georgia Institute of Technology

Institute for Human and Machine Cognition
Atlanta, GA

Marco Carvalho, Thomas C. Eskridge

Florida Institute of Technology

Institute for Human and Machine Cognition
Melbourne, FL

Abstract—Although a wide variety of multi-agent systems have been developed to address various processing-specific issues, they share a number of attributes concerning visualization and control. A set of minimum functionality is proposed that may be considered when designing visualizations representing multi-agent system properties and control options. Each of these guidelines helps to define a standard by which MAS visualizations might be designed to ensure clarity, regardless of specialization. These guidelines have been implemented in a lightweight and extensible visualization platform developed for an agent infrastructure called MIRA. We demonstrate several aspects of the this prototype, and discuss next steps for the system.

Keywords—*visualization; multi-agent system*

I. VISUALIZATION REQUIREMENTS

Although a multi-agent system (MAS) may be tailored to specific applications and thereby warrant more specialized visualizations, depicting agent presence, connectivity, general activity, and history is essential to interpreting the system's most fundamental attributes. An adherence to the following requirements allows for the development of a sufficiently articulate MAS visualization.

A. Depiction of High-Level Behavior

In dealing with abundant and multifaceted data, depicting more information at a lower granularity is a customary tradeoff for communicating an expansive performance model. Particularly in an MAS, this macroscopic approach helps to efficiently reveal emergent behavioral properties that would likely otherwise be overlooked by agent-specific observation. It also presents a context broad enough that it can be applied to agent systems of various types.

B. Depiction of Networked Events

Networked events can be defined both as communications (instances of agent to agent messaging) and relationships (hierarchical and associative data). In order to comprehend MAS processes, these constitutional properties must be illustrated effectively, but not so explicitly depicted that the model becomes obfuscated and uninterpretable. Acknowledging this balance, filters, whether computational or visual, must be employed to censor redundant or insignificant occurrences.

C. Depiction of Task Decomposition

MAS analysis requires transparency and traceability so that the origins and consequences of potential discrepancies might be discovered. Accordingly, it is useful to define the conditions that constitute a faulty circumstance: what distinguishes between appropriate and erroneous behavior, and is the answer applicable to all agent systems? Although there is no sure method of highlighting discrepancies as they arise, the visualization should suggest lapses in MAS logic and performance, whereby the user can investigate those of interest.

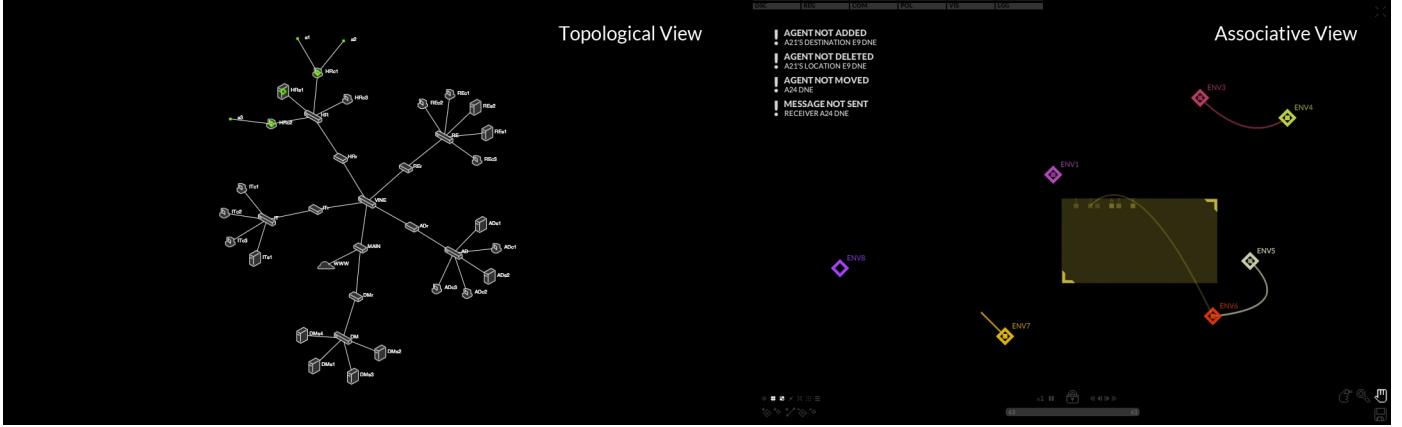
D. Depiction of History

Thorough MAS analysis requires that temporal qualities are considered, as changes over time are characteristic of all agent systems. So as to compensate for shortcomings in human memory, it should also be an aim of the visualization to represent all historical events simultaneously.

These requirements do not necessitate each property appearing explicitly on the screen; instead, the visualization design should consolidate these factors into intuitively comprehensible packages. For instance, it may be that properties are not represented by the explicit rendering of specific graphical elements (as in [8]), but are instead inferred from the holistic visual and/or behavioral composition of the display.

II. SOLUTION: MINIMALISTIC SEMANTICS

Design for the MIRA visualization began with developing semantic representations of agent interactions and their corresponding environments. Logically, environments are depicted as containers, and messages among agents appear as mediating wires. Although metaphoric entailments help to describe additional relevant properties (e.g. connection thickness is a function of message size and duration), little information is immediately displayed about the specific characteristics of each agent and environment. Instead, by forgoing the use of abundant textual elements and relying on minimalistic iconography, the broadest emergent features of the agent system are displayed with enhanced clarity. Resolution is sacrificed for transparency, and the result is a macroscopic view of the entire agent system, ideal for observing data where the locations of discrepancies are unknown. This all-inclusive approach differs from agent visualizations that depict entities and their relationships as minified state machines, as in [4] and [7].



In order to display the visualization as a cohesive whole, GUI elements such as tools and viewing options are not isolated to individual panels or windows as in [3], but rather consolidated into the principal display. Still, these static toolkit options remain disassociated from the visualization's dynamic content: each is spatially fixed and aesthetically distinct from the remainder of the display. As a result, agent and environment entities along with their emergent forms appear malleable and become the apparent focus of user observation and interaction, whereas the peripheral elements resemble a heads-up display, understood as alternative modifiers and constants throughout visualization changes.

III. SOLUTION: CONTEXTUALIZATION OPTIONS

When the visualization's macroscopic perspective is made intuitively comprehensible, the user may then situate the model in different observation-yielding contexts, effectively querying for higher resolutions of data. For example, as opposed to individually observing a number of environments and noticing that an assorted few are running on interconnected devices, the user might choose to display the environments on a diagram of network connectivity. Particularly in developing distributed systems, allowing the user to adjust the model *in context* gives them the capability to isolate potential problem areas first, and consequently expose faults in logic and performance more easily than sweeping computational analysis, like fault propagation simulation in [7]. Shifting the existing visual model about a variety of configurations provides a methodological, as opposed to a trial-and-error, approach to identifying discrepancies and behaviors.

An additional affordance of this functionality is that the visualization is more readily extensible: additional properties of data may be represented by custom-tailored distortions of the default model. Rather than overloading the display with textual and graphical indicators, an alternative perspective may simply be made available upon developer definition. A series of perspectives has been developed to represent instances of this extensibility - each takes the same information about agents and their behavior and interprets it to different visual effects. Noteworthy amongst these configurations are an associative perspective, wherein agent environments are clustered by communication, a topological perspective where agents are situated on a diagram of network connectivity, and a

chronological perspective that adjusts agent objects on a horizontal plane depending on their activities.

IV. CONCLUSION

MIRA's visualization, in offering a wide variety of customization options, struggles to provide a consistent conceptual model throughout its displays. Despite involving shared instantial elements, the visualization perspectives are not easily compared nor observed in tandem. Establishing a loose semantics across configurations helps to make sense of their functions, but may be insufficient in efficiently drawing conclusions between patterns.

Nevertheless, the MIRA visualization platform has proven itself as a useful framework for developing the MIRA infrastructure for which it was designed. Furthermore, its emphasis on extensibility suggests a model for developing visualizations that do not simply translate MAS data, but act as liaisons between the user and the agents' aggregate behavior. As a result of ensuring only that the most intrinsic properties of MASes are depicted, the platform yields visualizations just intricate enough to describe features pertinent to all agent development scenarios.

REFERENCES

- [1] A. Agogino and K. Tumer, "Analyzing and Visualizing Multi-Agent Rewards in Dynamic and Stochastic Domains," *Autonomous Agents and Multi-Agent Syst.*, vol. 17 i. 2, pp. 320-338, Oct. 2008.
- [2] M. Kögler, "Simulation and Visualization of Agents in 3D Environments," Diploma thesis, Univ. Koblenz-Landau, Germany, 2003.
- [3] M. Liedekerke and N. Avouris, "Debugging Multi-Agent Syst.," *Information and Software Technology*, vol. 37 no. 2, pp. 103-112, 1995.
- [4] D. Ndumu *et al.*, "Visualising and Debugging Distributed Multi-Agent Syst." in *Proc. 3rd Annu. Conference on Autonomous Agents*, 1999 © ACM. doi: 10.1145/301136.301220
- [5] S. Panzieri *et al.*, "An Agent Based Simulator for Critical Interdependent Infrastructures" in *Securing Critical Infrastructures*, 2004.
- [6] H. Parunak, "Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Anal." in *Proc. 2nd International Conference on Multiagent Syst.*, 1996 © AAAI.
- [7] D. Poutakidis *et al.*, "Debugging Multi-Agent Syst. Using Design Artifacts: The Case of Interaction Protocols" in *Proc. AAMAS '02*, 2002 © ACM. doi: 10.1145/544862.544966
- [8] G. Taylor *et al.*, "VISTA: A Generic Toolkit for Visualizing Agent Behavior" in *Proc. 11th Conference on Comput. Generated Forces and Behavioral Representation*, 2002.