MAS Infrastructure
Definitions, Needs, and Prospects

Les Gasser

Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
501 East Daniel St., Champaign, IL 61820
gasser@uiuc.edu
WWW: http://www.uiuc.edu/~gasser

Abstract. This paper attempts to articulate the general role of infrastructure for multi-agent systems (MAS), and why infrastructure is a particularly critical issue if we are to increase the visibility and impact of multi-agent systems as a universal technology and solution. Second, it presents my current thinking on the socio-technical content of the needed infrastructure in four different corners of the multi-agent systems world: science, education, application, and use.

1 Why MAS Infrastructure is an Issue

MAS have the potential to meet two critical near-term needs accompanying the widespread adoption of high-speed, mission-critical content-rich, distributed information systems. First, they can become a fundamental enabling technology, especially in situations where mutual interdependencies, dynamic environments, uncertainty, and sophisticated control play a role. Second, they can provide robust representational theories and very direct modeling technologies to help us understand large, multi-participant, multi-perspective aggregates such as social systems, ecologies, and large information processing systems. Many people inside and outside the MAS community can now legitimately envision a future in which we clearly understand how information and activity of all kinds can be managed by (automated) teams and groups (not individuals), and in which we naturally and ubiquitously manage it that way: a vision of "MAS everywhere."

Progress toward systematic scientific principles and robust coordination/interaction technologies for MAS has been underway for the past thirty years. Though more fully articulated knowledge is needed, we are on the way toward developing the knowledge that will eventually give MAS a comprehensible. predictable operational character. MAS researchers have developed some fairly sophisticated theories and technologies for multi-agent interaction; coordination; coalition formation; conflict resolution; dynamic organization and reorganization; network and organization design; fault-tolerance, survivability, and robustness; multi-agent learning; and real-time multi-agent behavior. In a theoretical sense, there is much interesting work and many good results, many of which point the way to more intriguing questions. It is fair to say that in several of these areas---coordination, teamwork, coalition-formation, dynamic reorganization, for example---the approaches developed in the multidisciplinary MAS community are among the most detailed, sophisticated and general that are available.

But from a practical point of view, our understanding is really just beginning. A number of deep scientific issues (such as managing dynamic heterogeneity [11] and understanding system-wide pathologies [10]) are very under-explored and have implications that will only arise clearly when we begin to have widespread, interacting, fielded MAS. Currently there is a very small number of fielded MAS systems, and in general there are very few---if any---systems in routine operational use that actively exploit the most sophisticated MAS theoretical developments such as robust inter-agent coordination techniques, multi-issue negotiations, dynamic organizational efficiencies, or multi-agent learning.

Moreover, even if the next generation of MAS technical milestones are met and new capabilities are created by researchers, widespread use of MAS won't occur until a critical mass of fielded systems, services, and components exists and network effects take hold to blossom the user population and public interest. The public incentives for widespread attention to and use of analogous technologies such as Web browsers and cell phones appeared only with the development of a) a stable, reliable, accessible infrastructures, and b) a critical mass of "content" (e.g., broadly interesting websites) that compelled potential users. Similarly, until we have a stable, operational, widely accessible, and low-apparent-cost MAS infrastructure populated with a critical mass of MAS services and systems that provide value to prospective users, MAS is likely to languish as a technology with potential, not kinetic, energy.

Another critical impact of the failure to have a variety of fielded MAS is that we lack practice and experience in building and operating MAS in situ. In virtually every case of implemented experimental or commercial MAS, the theoretical and technological frameworks used rely on standard, homogeneous agents and limited, inflexible standards of interactivity. Each project or application is generally self-contained and its components can accommodate only a very limited, predictable range of interaction. In the research community, each group's projects are quite often similarly isolated. Though there are some widely distributed MAS tools (see e.g., [1]) it is rare that one group's tools and technologies work with those of others in an integrated way, and cross-group testing generally doesn't happen. (However, see [15],[18].)

An important exception to this is recent experiments in constructing and using joint simulation environments and shared physical environment, such as the RoboCup simulations. Still, under these conditions there is generally still careful centralized control over interaction possibilities, determined for example by simulator APIs or controlled physical environments. Also worth mentioning are the newly emerging infrastructure tools, such as the Nortel FIPA-OS implementation, many attempts at KQML tools, and many XML frameworks e.g., for e-commerce [20]. It's not yet clear to what extent these will actually serve to integrate agent behaviors (see e.g., [11],[19].) The bottom line is that despite the compelling vision of ubiquitous multi-agent technology, we simply have hardly any real experience building truly heterogeneous, realistically coordinated multi-agent systems that work together, and thus almost no basis for systematic reflection and analysis of that experience.

Finally, the current prospects for advanced pedagogy in MAS are very weak, especially in terms of demonstration of MAS and experimentation in MAS behavior and implementation. How will the MAS communities create pedagogical environments and tools that will help develop, transfer, and extend the MAS knowledge and skills to impact widening groups of people? Simply put, there are few if any sharable tools with serious pedagogical aims.

2 MAS Infrastructure Elements and Attributes

An infrastructure is a technical and social substrate that stabilizes and rapidly enables instrumental (domain-centric, intentional) activity in a given domain. Said another way, (technical) infrastructure solves typical, costly, commonly-accepted community (technical) problems in systematic and appropriate ways. In this way, infrastructure allows much greater community attention to unique, domain-specific activities. As Star and Ruhleder have pointed out [16] infrastructures have the general character of being: embedded "inside" other structures; transparent (not needing reinvention or re-assembly each time); of wide reach or scope; learned as part of community membership; linked to conventions and norms of community practice; embodying standards, shaped by pre-existing installed bases of practice and technology; and invisible in use yet highly visible upon breakdown. Infrastructure is also an effective leveling device: it unifies local practices with global ones, both providing coordination and creating shared knowledge. For the purposes of this paper, I've divided spheres of MAS activity into four categories, each of which has different infrastructure needs---that is to say, the communities in each sphere have different views of their own "typical, costly, commonly-accepted community technical problems" and different notions of what are the most "systematic and appropriate" solutions to them. These four categories are MAS science, MAS education, MAS application, and MAS use. The most critical infrastructure needs are not the same across these focus areas, and not all of these area are developing with equal force or speed. (E.g., MAS science is way ahead of MAS use, and MAS application is somewhat ahead of MAS education). Table 1 presents a schematic view of MAS infrastructure elements and characteristics, and their relationship to each of the four MAS spheres. I'll treat each of these in more detail below.

3 Needed Elements of MAS Infrastructure

Rows of Table 1 show the main elements of MAS infrastructure, categorized into System Elements, Services, Capabilities, and Attributes. Table 1 shows the relationships of these infrastructure elements to needs in the four different categories of MAS activity.

MAS Infrastructure Needs	MAS Science	MAS Education	MAS Application	MAS Use
System Elements				
Communication Languages	E	E	E	N
Components (content and processes)	E	E	E	E
Comprehensive, Implemented MAS	E	E	E	E
Design Methodologies	D	D	E	N
Experimental Platforms	E	E	D	N
IDEs	E	D	E	N
Implementation Frameworks	E	E	E	N
Active Services				
Certification Services	D	E	E	E
Economic Services	P	E	E	E
RDD Services	E	E	E	E
Security Services	D	E	E	E
Specialized Domain Services	D	D	E	E
Capabilities				
Analysis	E	D	P	N
Data Collection	E	E	P	P
Experiment Construction	E	E	D	N
Information Exchange	E	E	P	P
Intentional Failure	E	D	P	N
Measurement	E	E	E	N
Representation of MAS Concepts/Data	E	D	E	D
Simulaton	E	E	D	N
Transfer	E	D	E	D
Attributes (of Elements/Services/Capabilities				
Administrative/Economic Practicality	E	E	E	E
Illustrativeness	D	E	P	N
Openness	E	D	E	P
Packaging	D	E	E	E
Progressive Complexity	D	E	E	P
Robustness	D	E	E	E
Scalability(many dimensions of scalability)	E	P	E	E
Sharablility	E	E	D	P
Standardization	D	D	D	D
Support	P	D	E	E
Usability	E	E	E	E
Visibility	E	E	P	P
Widespread Availability	D	E	E	E
Other				
Community	E	E	D	P
Open Source Projects	E	E	E	E
User Groups and Interest Groups	D	P	E	E

E = Essential
D = Desirable
P = Possibly useful
N = Not Critical

Table 1

The two primary categories of infrastructure elements are System Elements and MAS Services. System Elements are tool-level MAS needs important for actually constructing MAS. Services, both generic and specialized, refer to active online services needed for effective integration of MAS in one or more of the four "spheres." Capabilities refer to operational capabilities provided by MAS elements or services. Attributes describe ideal, valuable MAS-relevant characteristics of one or more of the elements and services. Finally, there are several infrastructure issues that fall outside this framework. Below, "MAS Infrastructure" is usually abbreviated to "MASI."

3.1 System Elements

Communication Languages: Support for Agent Communication Languages (ACLs) and their underlying support bases (e.g. belief knowledgebases for MAS based on KQML, FIPA, FLBC, etc.).

Components (content and processes): There should be extensible, easily used libraries of components that provide multi-agent oriented capabilities and tools and there should be a clear understanding of the relationships between agents, agent components, and implementation technologies such as distributed objects [8].

Comprehensive, Implemented MAS: Complete, operational MAS in particular, useful, significant domain areas that are available for experimentation and extension.

Design Methodologies: Systematic engineering methods for the design and construction of MAS, that increase productivity and integrity of the resulting products.

Experiment Platforms: Simulation and representation frameworks that are specifically oriented toward systematically experimenting with, developing, and testing multi-agent concepts, theories, and implementation approaches ([2],[3],[7],[9],

IDEs: Integrated Development Environments specialized for construction, operation and use of MAS.

Implementation Frameworks: Implementation Frameworks (IFs) are implemented, sharable architectural templates that can be filled in with specific MAS codes and data for applications. Examples include some existing agent-building toolkits with multi-agent capabilities [1]. IFs must capture multi-level models; models of the MAS operational environment and infrastructure (e.g. hardware models; failure), message communications; agents and their activities; tasks and problem-level interactions and dependencies (e.g., GPGP and TAEMS [2],[4]); agent ensembles and their interactions; possibly environmental models, such as physical landscapes or structures (these are notoriously hard).

The issue of the relative utility of alternative multi-agent architectures is an important one [14]. The lowest-level implementation frameworks should enforce no fixed agent architecture beyond message passing, execution, tracing, and some database capability. Above this, there should be infrastructural support for a set of multi-level, complex, sophisticated architectural templates to rapidly instantiate agents with specific architectures, knowledge, and policies, such as BDI agents and blackboard agents. Actually a variety of such templates is desired, ranging across the following:

-- Simple agents with simple message passing facilities (send-message) and simple, "flat" (non-hierarchical) reactive or programmatic decision models.

-- Agents with flat structure, some significant communication support such as structured Agent Communication Languages (ACLs) and simple procedural or reactive reasoning.

-- Agents with flat structure but some significant communication (e.g. structured ACLs) and knowledge-based reasoning such as JESS, Prolog, etc.

-- Agents with some significant communication support such as structured ACLs [6], XML [20], or GPGP [4], and with sophisticated, multi-level, multi-component reasoning/control architectures such as BDI, sophisticated blackboard control architectures; reasoning with a variety of time horizons such as organization-level and problem-level (e.g. planning) horizons, etc.

3.2 Active Services

Online, continuously running or demand- callable infrastructures or servers with a growing set of standard agents with general expertise in various tasks. These would provide the basis for an ongoing, growing, sharable agent service infrastructure.

Certification Services: Third-party security services that certify the origin or the security of an MAS and its components.

Economics Services: Services for charging and managing economic interactions in MAS.

Resource Description/Discovery (RDD) Services: Services for real-time and non-real-time description, offering, and discovery of MAS resources such as active teams; contracting partners; new markets; etc. (Also called Matchmaking services [5],[13],[17].)

Security Services: Services that create and enforce trust, truth-telling, and system integrity, including protecting property and other rights.

Specialized Domain Services: Specialized MAS services for particular operational domains that are available on an interactive basis, possibly subject to RDD, certification, economic, and security processes.

3.3 Capabilities

Analysis: Ability to effectively make sense of MAS data and information to synthesize new principles and to verify known ones.

Data Collection: User-defined probes for gathering data on states and events at level of:

-- operational infrastructure (e.g., message delivery; agent execution; size of agent queues)

-- agents (e.g., agent behavior; size/quality of agent database)

-- problem or task (e.g. goal creation, task assignment, goal mix, other "metadata"

Experiment Construction: Large-scale experiments involving large, heterogeneous MAS in differing local circumstances are very hard to set up, initiate, deploy and run under varying operational parameters [9]. Tools and support are needed for this, including batch modes for multiple runs (e.g., overnight Monte-Carlo simulations). Support (i.e., tools and languages) for easily specifying and constructing large MAS models is also needed. "Large MAS models" means large (100 to 100,000) ensembles of possibly sophisticated, possibly simple heterogeneous agents with heterogeneous relationships and connections, to significant collections of information and knowledge content, possibly via external content-bases. This is necessary to start and run experiments of realistic and interesting scale and scope.

Information Exchange: Open exchange of MAS information through written articles and reports; direct contact, and open availability of core materials such as source code

Intentional Failure: Intentional failure of agents must be supported for experimental purposes, Other failures that should be modeled and simulated include failure of infrastructure (e.g., communications) and communication delays.

Measurement: The ability to operationalize, capture, and measure aspects of MAS performance at a number of levels.

Representation of MAS Concepts/Data: Easy capture and representation of MAS concepts and data, e.g., teams and groups; distributed knowledge; distributed interpretation.

Simulation: The ability to run controlled (repeatable) simulations and to gather a wide variety of data from them is critical. To avoid behavioral artifacts of synchronous simulation, controlled randomization of individual agent actions and agent interactions, and explicit agent execution ordering are also necessary. These should incorporate MVC (model-view-controller) paradigms for realtime control/influence and observation of simulations. Serious multi-agent simulations trade off capability and complexity: large-grained agent architectures with sophisticated control reasoning use large amounts of simulation resources, and cry out for distributed approaches. However, distributed simulation techniques are very complicated, the moreso in conditions where inter-agent dependencies are emergent and not definable until runtime [9].

Transfer: Ability to transfer MAS and MAS components/elements to entirely new operating and community substrates with no loss of functionality. Ability to "unplug" agents and attach them to other systems or environments with simple runtime support or wrappers. This includes technology transfer of MAS infrastructure ideas.

3.4 Attributes of Elements/Services/Capabilities

Administrative/Economic Practicality: Good fit between resource needs and administrative requirements of the MAS and availability of same in the community of participants [12].

Illustrativeness: The ability to use the MAS to illustrate a principle or phenomenon by controlling its operating parameters and/or its execution process, and by offering support for visualizing or otherwise communicating results meaningfully. The ability to capture pictures and data streams for analysis and for papers and articles is also useful.

Openness: The ability to incorporate agents that are heterogeneous on many dimensions (architecture, resource used; interactivity; scale), possibly except for minimal 'wrapper', protocol, or API technology.

Packaging: Available as a complete self-contained package with supporting documentation and needed resources [12].

Progressive Complexity: The ability to progressively increase or decrease some aspect of the complexity of the MAS or its tasks or environments. This is useful for experimentation (variance), pedagogy (showing effects of increasing/decreasing complexity), and application development (prototyping, robustness testing). A collection of progressively more complex pedagogical agents and agent ensembles that illustrate important principles in MAS would be very useful. Ideally there would be some agents in such a collection that would be component-based, and for which components would be incrementally aggregatable. Thus students and teachers could explore alternative architectures and differentially complex agents, and could easily experiment with how added capabilities give added sophistication in multi-agent behavior.

Robustness: Continued operation of the MAS over a wide range of operating conditions and environments; Failure-tolerance and soft (progressive rather than precipitous) failure.

Scalablility (many dimensions of scalability): In terms of support for numbers of agents, MASI elements should support a range of from one or two agents to at least a hundred thousand. In the MASI Science realm, this is in part because the study of emergent large scale phenomena may require orders of magnitude different scale to exhibit the changes in phenomena.

Sharability: For community use, the MASI needs to be easily usable, easily comprehensible, and easily implementable/runnable in a variety of settings. It also must be sharable administratively.

Standardization: Conforming to community-wide standards for some dimension(s) of operation or interaction; Common, widely-used programming and agent-building languages such as Java. In many (but certainly not all) cases, MAS infrastructure will benefit from standardization of components, architectures, languages, interfaces, etc. [6]. Managing heterogeneity is an active area of MAS research [11], and we should be sensitive to alternatives to standardization that can provide both robustness and adaptability in addition to integration and sharability.

Support: Existence of some party responsible for extensions, modifications, upgrades; responsive to changes in its own infrastructure [12].

Usability: A high degree of correspondence between the skills, knowledge, resources, and organizational context of users and those required for effective use of the MAS.

Visibility: The ability for builders and/or users to access and to visualize internal dimensions, processes, interactions, and architectures of MAS in meaningful ways.

Widespread Availability: The degree of accessibility to and usability by a wide range of MAS participants and communities, including shared ownership, open source access, free modification/extension, Ease of location, retrieval, setup, and operation.

3.5 Other

Community: Thriving, communicative, responsive community ("community of practice") surrounding MAS.

Open Source Projects: MAS projects with protected free availability; open rights to modify, and complete representations (e.g., source code and documentation).

User Groups and Interest Groups: Communities of users and participants who actively share resources and knowledge to refine mutual understanding of MAS issues and to solve problems are actually infrastructure.

4 Conclusions

A vision of "MAS everywhere" means being strategic about infrastructure. Infrastructure needs for MAS are not uniform, and there are several constituent MAS communities that are important for progress in research, development, and application. Infrastructure is much more than specific abilities for standardized communication and resource discovery, and principles from other successful technologies should be investigated and used for inspiration in the MAS case.

5 Acknowledgements

My recent thinking on the issues discussed in this paper has benefited greatly from conversations with (alphabetically) Jean-Pierre Briot, Zahia Guessoum, Larry Jackson, Besiki Stvilia, Jun Wang, and others in the UIUC WEDAM and Agents and Multi-Agent Systems (AMAG) research groups. Tom Wagner’s perspective on the issues and his support in shepherding the paper have also been immensely helpful. This work has been supported in part under NSF Grant 9981797.

References

[1] AgentBuilder, Inc, "Agent Construction Tools" http://www.agentbuilder.com/AgentTools/, 2/2001

[2] Keith S. Decker. TAEMS: A framework for analysis and design of coordination mechanisms. In G. O'Hare and N. Jennings, editors, Foundations of Distributed Artificial Intelligence. Wiley Inter-Science, 1995.

[3] Keith S. Decker, “Distributed Artificial Intelligence Testbeds.” pp. 199-138 in N.R. Jennings and G.M.P. O’Hare (eds.), Foundations of Distributed Artificial Intelligence, John Wiley and Sons, New York, 1996.

[4] Keith S. Decker and Victor R. Lesser. "Generalizing the Partial Global Planning Algorithm." International Journal of Intelligent and Cooperative Information Systems, 1(2):319--346, June 1992.

[5] Keith Decker, Katia Sycara, and Mike Williamson, "Middle-Agents for the Internet." In Proceedings of the 15th International Joint Conference on Artificial Intelligence, Nagoya, Japan, August 1997.

[6] K. Decker, T. Finin, C. Manning, M. Singh, and J. Treur. "Implementing multi-agent systems: languages, frameworks, and standards (a report from the 1997 IW-MAS workshop)." AAAI-98 Workshop on Software Tools for Developing Agents, Technical Report, Computer Science Department, University of Delaware, AAAI-TR WS-98-10, August 1998.

[7] Les Gasser, Carl Braganza, and Nava Herman, "MACE: A Flexible Testbed for Multi-Agent Systems Research", in M.N. Huhns, Ed. Distributed Artificial Intelligence, Pitman, 1987.

[8] Les Gasser and Jean-Pierre Briot, "Agents and Concurrent Objects", IEEE Concurrency, October-December, 1998.

[9] Les Gasser, "Design Considerations for the MACE3J Simulation Environment", Working Paper LG-2001-02, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, February, 2001

[10] Mark Klein, "Exception Handling in Agent Systems." Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, 1999.

[11] A. Ouksel, A. Sheth (Eds.), Semantic Interoperability in Global Information Systems, Special Issue of ACM SIGMOD Record, Vol. 28, No. 1, March 1999, pp. 47-53.

[12] H. Van Dyke Parunak, "Industrial and Practical Applications of DAI", Chapter 9 in G, Weiss, (Ed.), Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, MA., 1999.

[13] Onn Shehory, "A Scalable Agent Location Mechanism," in Lecture Notes in Artificial Intelligence, Intelligent Agents VI, Springer, 1999.

[14] Onn Shehory, "Architectural Properties of Multi-Agent Systems," Technical Report CMU-RI-TR-98-28, Robotics Institute, Carnegie Mellon University, December, 1998. (http://www.ri.cmu.edu/pubs/pub_479.html)

[15] Onn Shehory and Katia Sycara "The Retsina Communicator." In Proceedings of Autonomous Agents 2000, Poster Session.

[16] S.L. Star and K. Ruhleder, "Steps to an Ecology of Infrastructure." Information Systems Research, 7(1), March, 1996.

[17] Sycara, K., Klusch, M., Widoff, S. and Lu, J. "Dynamic Service Matchmaking Among Agents in Open Information Environments". In [Ouksel and Sheth, 1999].

[18] Tambe, M., Pynadath, D. and Chauvat, N., "Building Dynamic Agent Organizations in Cyberspace," IEEE Internet Computing, Vol 4, number 2, March, 2000.

[19] Wooldridge, Michael, "Semantic Issues in the Verification of Agent Communication Languages." Autonomous Agents and Multi-Agent Systems, 3(1), March, 2000.

[20] The XML Catalog, XML.Org, http://xml.org/xmlorg_registry/index.shtml