Macro-Programming Multi-Agent Systems: A Framework for Artificial Collective Intelligence

Blue Sky Ideas Track

Roberto Casadei
Alma Mater Studiorum—Università di Bologna
Cesena, Italy
roby.casadei@unibo.it

ABSTRACT

This paper elaborates on the opportunity and idea of multi-agent system (MAS) macro-programming, i.e., programming in terms of macroscopic denotations of system goals, structure, or behaviour. Indeed, macro-descriptions with explicit macro-to-micro mapping can be a formidable way to harness the complexity of emergent collective behaviour (for humans) and to provide a structure for guiding optimisation, learning, and generative artificial intelligence (AI) processes (for computers). Despite contributions about meso-level (e.g., organisational), multi-level (e.g., holonic) and declarative (e.g., goal-oriented, normative) paradigms exist in the MAS literature, research is fragmented and the topic arguably overlooked by both the scientific and software engineering viewpoints. Recent survey works on macro-programming spanning areas from sensor networks to swarm robotics suggest that a synthesis is possible, despite the variety of methods, techniques, and abstractions. With reference to early and recent literature both within and outside the MAS community, this paper motivates that multi-scale and especially macroscopic descriptions are possible, useful, and timely—opening up to research opportunities and community debate.

KEYWORDS

multi-agent systems programming; artificial collective intelligence; macro-programming; multi-level models; micro-macro link

ACM Reference Format:
Roberto Casadei. 2026. Macro-Programming Multi-Agent Systems: A Framework for Artificial Collective Intelligence: Blue Sky Ideas Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 6 pages. https://doi.org/10.65109/SAXG2906

1 INTRODUCTION

Effectively engineering the *collective behaviour* [16] of a collection of interacting agents is a long-standing challenge [12, 13, 18, 57]. The essence of the issue lies in the *combinatorial* nature [54] and *complexity* of the combined system-environment dynamics, normally leading to *emergent* phenomena [86]. The design activity traditionally focusses on *individual agents' behaviours* and their *coordination/interaction* [11, 43, 58]. However, this classical approach is subject to the *forward*, *local-to-global*, or *micro-to-macro* prediction problem (foreseeing the overall behaviour and effects) [68, 78, 82], generally addressed by bottom-up, try-and-error design and assessed via (expensive) *simulation* of system behaviour in a (limited) range of environments. *Automatic approaches* based on search, optimisation, and machine learning have also been proposed. For instance, *evolutionary* [76] and *multi-agent reinforcement learning (MARL)* [39] have contributed by using supervisory signals to search through the policy space, but tend to suffer from issues like scalability, reward design, credit assignment, generalisability, adaptation, explainability [51, 88].

Over time, *language-based* proposals emerged for addressing the ***inverse, global-to-local* or *macro-to-micro* problem**, for specific problems like swarm task allocation [78] or spatial self-organisation [79]. Research of this kind has been quite fragmented across fields. Recently, attempts to synthesis were proposed along the umbrella term of ***"macro-programming"*** [17, 44] roughly meaning "programming by a macroscopic perspective" (cf. Figure 1).

Macro-programming as a term and abstract notion was first proposed in early 2000s for programming wireless sensor networks (WSNs) while abstracting communication and data routing [56]. A new wave then arose with the Internet of Things (IoT) [44] and swarm robotics [13]. Interestingly, however, while related MAS approaches like *normative* [21], *organisational programming* [40], and *holonic* MASs [8, 31] have been investigated, the idea of macro-programming has not really entered the MAS community.

To foster debate and research in the MAS community, this paper aims to brush the long-term ***micro-macro link design*** challenge [68] up and frame the idea of *MAS macro-programming*. Specifically, we provide an original discussion of macro-programming under the lenses of MAS research, based on related MAS literature [8, 21, 31, 40], recent advancements [4, 25, 34, 64], and recent attempts towards a synthesis [17, 18, 44]. To promote future investigations, we put together a variety of recent ideas and facilitators from multiple research communities (e.g., coarse-graining, downward causation, complexity magnifiers, etc.) into the macro-programming framework, and discuss key research opportunities and challenges for the MAS community.

2 A DISCUSSION OF MACRO-PROGRAMMING UNDER THE LENSES OF MAS RESEARCH

This section provides an original discussion of MAS macro-programming (Figure 1), inspired by work in [17, 44], offering a novel view on the long-term micro-macro challenge in

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/SAXG2906

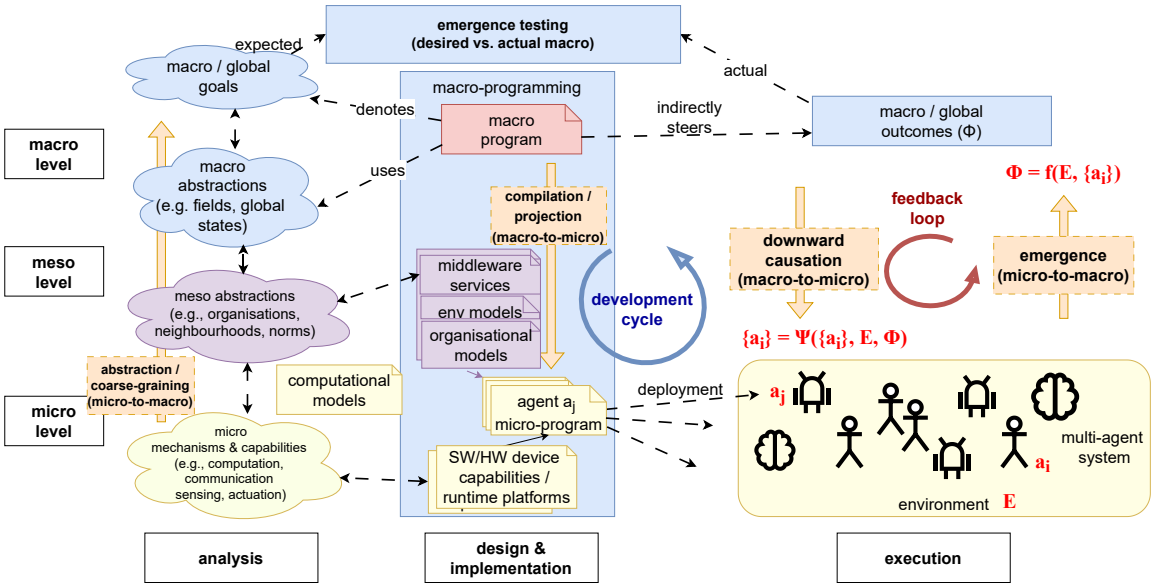

Figure 1: The idea of macro-programming MASs by abstraction level (y-axis) and software phase (x-axis).

MASs [68, 71]. The key point is the use and combination of *multi-level descriptions*, and especially *macro-descriptions*, for the engineering of MASs. We use term "description" to broadly cover both approximate representations of some system aspect or functionality (cf. prompts for generative AI (GenAI)/large language models (LLMs)), as well as formal specifications up to executable programs (*macro-programs* [17])—though we emphasise the latter.

Preliminaries: Scales, Scopes, and Levels. In MASs, it is common to distinguish the micro level of *individual agents* with the macro level of an agent *society* [83]. In general, many kinds of natural and artificial systems are intrinsically *multi-level (hierarchical)* and *multi-scale*. This is a result of using *abstraction* (removal of details) when describing a system in terms of related *entities*—be it for analysis or design. A *scale* denotes a granularity of observation (e.g.: micro, meso, macro): at the micro scale, we can observe fine-grained entities, whereas at the macro scale, we can only observe coarse-grained entities. A *scope* denotes the extension of an observation (e.g.: local, global): a local scope gathers close entities (cf. partial observability, and neighbourhoods), whereas a global scope gathers all entities. A *level* or *layer* refers to grouping abstractions by a unidimensional, ordered perspective. E.g., a system model may stack the *physical*, *software*, and *logical* levels. Often, levels are combined with scale: so, e.g., the micro level includes micro-scale entities; above, the meso level is stacked (e.g., groups of entities), with the macro level at the top (cf. global, coarse-grained states).

Preliminaries: Goals and MAS Types. We are interested in the engineering of *artificial collective intelligence* [16] by a computer science perspective: i.e., defining the control logic of a group of situated agents in a way that it enables them to solve problems in some environment beyond their individual capabilities. Multiple taxonomies of collective tasks [13, 32] and multi-agent systems [26, 42] exist, suggesting ways to position and set the scope of macro-programming. Regarding tasks, we consider complex, temporally extended collective tasks of different *nature* • *information-intensive* tasks [87], as in collaborative information processing, sociotechnical systems, and federate learning; • *spatiotemporal* tasks, as in pattern formation and collective transport, or • *mixed* tasks [13], with *coordination requirements* (multiple agents are needed to solve the task). Regarding the system to be macro-programmed, we assume it consists of *multiple* (often many), possibly *heterogeneous agents* (e.g., differing w.r.t. capabilities and computational model), which may interact with arbitrary means (e.g., message-passing or *stigmergy* [36]). Examples of existing and potential macro-applications may include swarm-based search-and-rescue [24], smart city operations (e.g., waste management [41]), games (e.g., soccer [50]), and human-AI collaborative workflows (e.g., software development [89]).

2.1 Macro-Programming for MASs

At the state of the art [17, 44], the idea is to consider macro-programming as a "high-level" programming paradigm [80] where programs (a.k.a. *macro-programs*) consist of *denotations* of teleological, structural, and/or behavioural aspects of a *collective* (homogeneous) or *composite* (heterogeneous) [18] of entities and agents.

Consider Figure 1, which depicts the macro-programming idea across two dimensions: *levels* (macro, meso, micro) and *development phases* (analysis, implementation, execution). Analysis aims to devise *macro/global goals*, using macro-abstractions which are based on meso- and micro-abstractions and may include, e.g., *fields* (maps from environment and agents to values) [45, 52, 79] and *global states* (functions of environment state and agent's states) [27]. Design and implementation involves turning system-wide goals into operational terms through a macro-program, which uses macro-abstractions. So, e.g., the programmer can explicitly denote and

manipulate an entire "swarm" of robots as a first-class program entity (cf., Buzz [63]), which may be useful to specify *missions* and team-wide tasking. Regarding this, an important remark is that a centralised program does not necessarily entail a centralised implementation: often, through *local projection* (e.g., by compilation, or contextual interpretation), it is possible to devise the agent controllers for decentralised execution. Also, one can define *macro-actions* like "follow leader", "uniformly cover that target area", "scatter to reduce the crowding level of agents", and so on. Another example is using a "field of movement vectors" to denote the collective movement of a swarm [3]. The essential characteristics of such instructions appear to be (i) *declarativity* (not specifying all the details) and (ii) *multi-directionality* (affecting multiple agents).

Indeed, a macro-programming solution typically combines a *language* for expressing macro-programs and some *middleware* to provide meso- or multi-level services: their interplay addresses the micro-macro link (via different approaches). During execution or simulation, macro-outcomes – functions of the environment state and agents' micro-states (also called *recombining factors* in [54]) – emerge from micro-level activity. It is key to observe that, through the so-called *downward causation* [86], macroscopic outcomes in turn affect micro-level states and behaviours (by so-called *necessitating factors* in [54]), giving rise to a feedback loop [8, 25].

2.1.1 Why Macro- and Multi-Level Descriptions. The motivation for using descriptions at multiple levels is that they fit different purposes or needs. E.g., in *multi-level modelling and simulation* [70], the granularity of models can be used to (i) achieve the desired *accuracy vs. efficiency tradeoffs*, by using a level of detail providing an acceptable approximation, and (ii) support *reusability* of models.

For engineers, instead, more *coarse-grained descriptions* can (i) promote understanding of system structures and intents [69], (ii) enable methodologies like *top-down design* [20], and (iii) defer or delegate the filling of implementation details at later stages and to other components. Indeed, in traditional bottom-up approaches focussing on individual agent behaviour, program and system dynamics comprehension are both undermined by the *gap* between the specified micro-behaviour and the resulting macro-behaviour (cf. emergence [86]). By a methodological viewpoint, focussing on the macro level fosters designing solutions starting from the global goals. Finally, by abstracting from lower-level details yet still providing a high-level guidance (hence constraining the behaviour space), macro-descriptions could represent a powerful bridge towards *automatic design* [13, 30], implemented via techniques like machine learning, optimisation, and program synthesis [33, 39, 76]—see [2] as an example of preliminary evidence on this possibility.

2.1.2 On the Possibility *of Macro-Descriptions.* Two recent surveys on macro-programming [17, 44] report on several proposals within 2000s-2020s for programming WSNs, robot swarms, and related systems abstracting from certain micro-level details (often, the message-passing logic used to collect and extract aggregated information). Different research groups have proposed distinct approaches with the claim that they somewhat address the macro-to-micro/global-to-local problem in specific settings or sub-problems [78, 79]. E.g., the *aggregate programming* approach [19, 79] has shown to enable macroscopic descriptions of self-organising spatio-temporal behaviour (e.g., distributed data processing via information flows [81], and collective movement of agents [3]) for connected and relatively dense groups of computing nodes.

2.1.3 The Micro-to-Macro Challenge: Operationalising Macro-Descriptions. The crucial point is *how* macro-structures and macro-actions are further *implemented*, i.e., mapped to local behaviours. This pertains to the design of a macro-programming solution. Though there is yet no comprehensive account on possible architectures, proposals include *local projection* as in choreographies [55], *progressive refinement* as in "domain objects" [14], *delegation and orchestration* as in collaborative process models [67], *contextual interpretation* as in aggregate programming [79], and *contextual adaptation* as in context-role oriented programming [34].

The organisation of the implementation support in terms of one or more *languages* (for specification) and *middleware/runtime layers* (for execution), and what consequences arise from a given *multi-level language-middleware interplay* on the programmer (e.g., in terms of intelligibility and analysis/design ability) is the very centre of any theory and principled practice (both currently missing) of macro-programming. We detail on this challenge in Section 3.

2.1.4 Related Notions in MAS Research. According to [17], macro-programming is founded on (i) micro-macro distinction (a cornerstone of the MAS community), and (ii) macro-to-micro mapping, i.e., emphasis is on inducing or deriving individual or local behaviour from macroscopic specifications. Thus, related work is at the intersection of *multi-level/collective/declarative* approaches.

Various notions of *multi-level descriptions* have been proposed in the MAS engineering literature [8, 31, 59, 62]. However, it is key to distinguish what is the target of the layering: (i) *Behaviour abstraction:* e.g., in [62], high-level abstract behaviours are denoted by *goal plans* that may reuse other goals or lower-level action plans (plans associated to no goals); here, the levels denote the abstraction of *individual* goals/actions. (ii) *Structure:* the *holonic* MAS paradigm [8, 31] considers *holons*, namely agents that can contain and/or be part of other agents, hence representing whole-part constructs and giving rise to hierarchical structures (*holarchies*) where levels are *explicitly* and uniformly described. (iii) *Operational semantics:* e.g., in [59], the authors propose a multi-level approach to the formal semantics of agent societies, based on *"count-as"* relations between actions at different levels, and aimed at allowing "vertical modularity" (reuse of semantic models across levels).

Various notions of *declarative* or *"global-level" descriptions* have also been proposed in the MAS engineering literature—e.g., the 2008 AAMAS paper by Yamins and Nagpal proposing"global-to-local programming" [84] of pattern formation in *spatial computers* [9]: this is actually the prelude of what became the already-mentioned *aggregate computing* paradigm [79]. Such approach is quite effective for expressing spatiotemporal self-organisation patterns, but seems hardly applicable to composites or heterogeneous MASs in general.

Another class is given by *declarative* approaches [7, 15], often logic-based, that specify *what* are the goals (rather than *how* they should be pursued). This also connects to the ***normative MASs*** paradigm [21], which aims to rule and constrain the behaviour of the agents through *norms* fostering the proper functioning of the whole MAS, especially in heterogeneous and open settings. Related techniques generally focus on the *meso-level* but may also cover the

macro-level (cf. global goals). In *organisational approaches* [40] and *organisational programming* with languages like Moise+ [11], there is support for the design of missions in terms of groups, roles, norms, and global goals decomposed based on social schemes. Another example is given by approaches that aim to specify the *global* collaborative logic of multiple agents (or parties): this includes *interaction protocols* with language like Kiko [77] and *choreographic programming* approaches [55]. Techniques where behaviours are generated or driven by (non-micro) norms [23] should be regarded as macro-programming and integrated into the taxonomies of [17, 44], distinguishing the *levels of autonomy* of individual agents.

3 RESEARCH DIRECTIONS

Operationalising the Scientific Discoveries on the Micro-Macro Link. The micro-macro link is studied in many research areas. E.g., the theoretical framework of *cellular automata* [10] provides a long-term record of studies about how local rules produce global behaviour. Often, simplified settings are assumed to prove theorems (e.g., reachability of desired states). However, rules are local and theorems tend to break when relaxing the assumptions. Aggregate programming adopts a computational model that is basically the same as cellular automata (while relaxing the assumptions and allowing for environmental changes, perturbations, etc.); still, quite surprisingly, there are no exchanges between those research silos. Another inspiring work is that on *computational mechanics* [66], which provides a formal setting for studying how information is stored and transmitted across scales. The notion of "computational closure" [66] characterises processes that are self-contained w.r.t. information-processing. This seems very much as a goal for macro-programs: being "effective theories" for the desired collective behaviours. The construct of *complexity magnifiers* [49] is also interesting, for it enables simple means to control complex emergent behaviour. A key research goal is connecting such measures and patterns to the realm of programming languages, to enable formal analysis and principled design. One key suggestion is to investigate on the bridge provided by formal ***collective computation models*** [6, 29, 35, 38, 46].

Explicit Multi-Scale System Analysis and Design Methodologies. Despite contributions exist [8, 34, 79], more research work is needed about addressing the micro-macro link in *explicit* terms. Also, there is very limited ***methodological guidance*** for the analysis and design of multi-scale systems (cf. [1] for collective MASs). By an analysis perspective, what *metrics* can be used to denote "how much macro" a program is, and the abstraction benefits? In general, how to evaluate the effectiveness of macro-descriptions for the purpose of supporting understanding and activities by human developer? Though arguments exist [18], *evidence-based* software engineering studies [60] addressing such questions are entirely missing. The development of *visual analytics (VA)* [72, 90] tools to make sense of multi-level relationships and effects is also desirable. Regarding design, what micro-to-macro and macro-to-micro patterns are available and when to use them in scenarios differing by heterogeneity, topology, knowledge representation (cf. symbolic vs. sub-symbolic), agency (cf. deliberative vs. reactive)? A deep understanding of *collective* and *composite* structures and dynamics [18, 53] and how they affect the engineering process is especially worthy. It naturally leads to the theme of *hybrid* and ***multi-paradigm*** *MAS design* [37, 61], which is also an urgent and timely direction—especially in this era where multiple types of agents and effective techniques for agent design exist (cf. BDI-[11], RL-, swarm-, and GenAI/LLM-based).

Emergence Testing and Runtime Verification. How to effectively *verify* emergent behaviour [48]? Generally, the idea is to run multiple *simulations* in multiple environment settings and dynamics. For instance, swarms of different sizes and topologies, perturbations of different kinds, and adversaries of different sophistication are used to understand the *working boundaries*. The problem is that simulations take time and resources, and the combinatorial nature of the possible environments and system dynamics make the *parameter* or *testing space* explode. Techniques for ***search-based test case generation*** [5] and ***smart sampling*** of the parameter space could prove useful, but these should also be combined with ***parameterised model checking*** [47], runtime verification [28], or formal proofs exploiting knowledge about the micro-macro link.

Macro-Programs for Guiding and Constraining Multi-Agent Learning. A key problem in MARL and evolutionary approaches with many agents involved is the combinatorial explosion of the search space [51]. Homogeneity can be exploited to induce more compact system representations [74]. When not possible, techniques for pruning the search space are key to support scalability. In this sense, hybrid approaches that combine behavioural specifications and learning/search algorithms may provide a viable strategy. The interesting notion of ***"macro-action"***, meant as a temporally-extended sequence of actions to be run as a whole, was proposed in the context of reinforcement learning a long time ago [65] and recently investigated in MARL settings [75]. Intuitively, this represents a potential bridge between macro-programming and MARL that must be investigated, together with ***sketch-based synthesis*** [2, 33, 73].

Generative AI / LLM for Addressing the Micro-Macro Link. Can generative AI and LLMs help us solve the micro-macro link in any way? Can they generate macro-programs for specific problems? In [4], Aguzzi et al. use prompts to let a LLM learn aggregate programming and then test its ability to solve collective computing problems. It works, but with severe limitations emerging quickly when rising the complexity or abstraction. Studies on the ability of LLMs to carry out ***multi-scale and multi-level reasoning*** [85] may prove useful here.

4 WRAP-UP

This paper offers a vision for addressing the micro-macro link in MASs using macro-abstractions and explicit multi-level designs. The goal is to provoke debate and stimulate the MAS community to integrate recent developments in "concrete" swarm- and macro-programming [17, 44, 63, 78] with long-standing MAS approaches (e.g., normative, holonic, choreographic) [8, 11, 31, 40, 55, 77] and recent scientific advances on collective computation [6, 29, 35, 79], emergence detection [10, 66, 86], and multi-agent AI [22, 39].

ACKNOWLEDGMENTS

This contribution is part of the broader research agenda of the *Italian Science Fund (FIS3)* Starting Grant project *FoMaSE - Foundations for Macro-programming-based Software Engineering* (Grant No. FIS-2024-00174, CUP J53C25002170001).

REFERENCES

[1] Dhaminda B. Abeywickrama, Nicola Bicocchi, Marco Mamei, and Franco Zambonelli. 2020. The SOTA approach to engineering collective adaptive systems. *Int. J. Softw. Tools Technol. Transf.* 22, 4 (2020), 399–415. https://doi.org/10.1007/S10009-020-00554-3

[2] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. 2022. Towards Reinforcement Learning-based Aggregate Computing. In *Coordination Models and Languages - 24th International Conference, COORDINATION (LNCS, Vol. 13271)*. Springer, 72–91. https://doi.org/10.1007/978-3-031-08143-9_5

[3] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. 2025. MacroSwarm: A Field-based Compositional Framework for Swarm Programming. *Log. Methods Comput. Sci.* 21, 3 (2025). https://doi.org/10.46298/LMCS-21(3:13)2025

[4] Gianluca Aguzzi, Nicolas Farabegoli, and Mirko Viroli. 2025. A Language-based Approach to Macroprogramming for IoT Systems through Large Language Models. *ACM Trans. Internet Things* 6, 4 (2025), 1–30. https://doi.org/10.1145/3758326

[5] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. 2017. Search-based test case generation for Cyber-Physical Systems. In *IEEE Congress on Evolutionary Computation (CEC)*. 688–697. https://doi.org/10.1109/CEC.2017.7969377

[6] Giorgio Audrito, Roberto Casadei, and Gianluca Torta. 2024. A general framework and decentralised algorithms for collective computational processes. *Future Gener. Comput. Syst.* 158 (2024), 11–27. https://doi.org/10.1016/J.FUTURE.2024.04.020

[7] Matteo Baldoni, Cristina Baroglio, Viviana Mascardi, Andrea Omicini, and Paolo Torroni. 2010. *Agents, Multi-Agent Systems and Declarative Programming: What, When, Where, Why, Who, How?* Springer, 204–230. https://doi.org/10.1007/978-3-642-14309-0_10

[8] Gillian Basso, Massimo Cossentino, Vincent Hilaire, Fabrice Lauri, Sebastian Rodriguez, and Valeria Seidita. 2016. Engineering multi-agent systems using feedback loops and holarchies. *Engineering Applications of Artificial Intelligence* 55 (2016), 14–25. https://doi.org/10.1016/j.engappai.2016.05.009

[9] Jacob Beal, Stefan Dulman, Kyle Usbeck, Mirko Viroli, and Nikolaus Correll. 2013. Organizing the Aggregate: Languages for Spatial Computing. In *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*. IGI Global, Chapter 16, 436–501. https://doi.org/10.4018/978-1-4666-2092-6.ch016

[10] Kamalika Bhattacharjee, Nazma Naskar, Souvik Roy, and Sukanta Das. 2018. A survey of cellular automata: types, dynamics, non-uniformity and applications. *Natural Computing* 19, 2 (2018), 433–461. https://doi.org/10.1007/s11047-018-9696-8

[11] Olivier Boissier, Rafael H Bordini, Jomi Hubner, and Alessandro Ricci. 2020. *Multi-agent oriented programming: programming multi-agent systems using JaCaMo*. MIT Press.

[12] Alan H. Bond and Les Gasser. 1988. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[13] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1 (2013), 1–41. https://doi.org/10.1007/S11721-012-0075-2

[14] Antonio Bucchiarone, Martina De Sanctis, and Marco Pistore. 2014. Domain Objects for Dynamic and Incremental Service Composition. In *Service-Oriented and Cloud Computing - 3rd European Conference, ESOCC (LNCS, Vol. 8745)*. Springer, 62–80. https://doi.org/10.1007/978-3-662-44879-3_5

[15] Roberta Calegari, Giovanni Ciatto, Viviana Mascardi, and Andrea Omicini. 2020. Logic-based technologies for multi-agent systems: a systematic literature review. *Auton. Agents Multi Agent Syst.* 35, 1 (2020). https://doi.org/10.1007/s10458-020-09478-3

[16] Roberto Casadei. 2023. Artificial Collective Intelligence Engineering: A Survey of Concepts and Perspectives. *Artif. Life* 29, 4 (2023), 433–467. https://doi.org/10.1162/ARTL_A_00408

[17] Roberto Casadei. 2023. Macroprogramming: Concepts, State of the Art, and Opportunities of Macroscopic Behaviour Modelling. *ACM Comput. Surv.* 55, 13s (2023), 275:1–275:37. https://doi.org/10.1145/3579353

[18] Roberto Casadei, Gianluca Aguzzi, Giorgio Audrito, Ferruccio Damiani, Danilo Pianini, Giordano Scarso, Gianluca Torta, and Mirko Viroli. 2025. Software Engineering for Collective Cyber-Physical Ecosystems. *ACM Trans. Softw. Eng. Methodol.* 34, 5 (2025), 153:1–153:40. https://doi.org/10.1145/3712004

[19] Roberto Casadei and Mirko Viroli. 2024. Declarative Macro-Programming of Collective Systems with Aggregate Computing: An Experience Report. In *26th International Symposium on Principles and Practice of Declarative Programming, PPDP*. ACM, 5:1–5:5. https://doi.org/10.1145/3678232.3678235

[20] Valentino Crespi, Aram Galstyan, and Kristina Lerman. 2005. Comparative analysis of top-down and bottom-up methodologies for multi-agent system design. In *4th International Joint Conference on Autonomous agents & multiagent systems (AAMAS05)*. ACM, 1159–1160. https://doi.org/10.1145/1082473.1082672

[21] N. Criado, E. Argente, and V. Botti. 2011. Open issues for normative multi-agent systems. *AI Commun.* 24, 3 (2011), 233–264. https://doi.org/10.3233/aic-2011-0502

[22] Kai Cui, Anam Tahir, Gizem Ekinci, Ahmed Elshamanhory, Yannick Eich, Mengguang Li, and Heinz Koeppl. 2022. A Survey on Large-Population Systems and Scalable Multi-Agent Reinforcement Learning. https://doi.org/10.48550/ARXIV.2209.03859

[23] Viviane Torres da Silva. 2008. From the specification to the implementation of norms: an automatic approach to generate rules from norms to govern the behavior of agents. *Auton. Agents Multi Agent Syst.* 17, 1 (2008), 113–155. https://doi.org/10.1007/s10458-008-9039-8

[24] Ulrich Dah-Achinanon, Seyed Ehsan Marjani Bajestani, Pierre-Yves Lajoie, and Giovanni Beltrame. 2023. Search and rescue with sparsely connected swarms. *Auton. Robots* 47, 7 (2023), 849–863. https://doi.org/10.1007/S10514-022-10080-7

[25] Ada Diaconescu, Louisa Jane Di Felice, Patricia Mellodge, and Payam Zahadat. 2025. Macro-Types in Multi-Scale Feedback Systems: A Conceptual Framework. In *IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. 64–75. https://doi.org/10.1109/ACSOS66086.2025.00023

[26] Gregory Dudek, MichaelR.M. Jenkin, Evangelos Milios, and David Wilkes. 1996. A taxonomy for multi-agent robotics. *Autonomous Robots* 3, 4 (1996). https://doi.org/10.1007/bf00240651

[27] R. Fagin, J.Y. Halpern, Y. Moses, and M. Vardi. 2004. *Reasoning About Knowledge*. MIT Press. https://books.google.it/books?id=xHmlRamoszMC

[28] Angelo Ferrando and Vadim Malvone. 2022. *Towards the Combination of Model Checking and Runtime Verification on Multi-agent Systems*. Springer International Publishing, 140–152. https://doi.org/10.1007/978-3-031-18192-4_12

[29] Jessica C. Flack. 2017. Coarse-graining as a downward causation mechanism. *Philos Trans A Math Phys Eng Sci* 375, 2109 (2017), 20160338. https://doi.org/10.1098/rsta.2016.0338

[30] Gianpiero Francesca, Manuele Brambilla, Arne Brutschy, Vito Trianni, and Mauro Birattari. 2014. AutoMoDe: A novel approach to the automatic design of control software for robot swarms. *Swarm Intell.* 8, 2 (2014), 89–112. https://doi.org/10.1007/S11721-014-0092-4

[31] Christian Gerber, Jörg Siekmann, and Gero Vierke. 1999. *Holonic multi-agent systems*. Technical Report. https://doi.org/10.22028/D291-24979

[32] Brian P. Gerkey and Maja J. Matarić. 2004. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robotics Res.* 23, 9 (2004), 939–954. https://doi.org/10.1177/0278364904045564

[33] Sumit Gulwani, Oleksandr Polozov, and Rishabh Singh. 2017. Program Synthesis. *Foundations and Trends® in Programming Languages* 4, 1–2 (2017), 1–119. https://doi.org/10.1561/2500000010

[34] Christian Gutsche, Sebastian Götz, Volodymyr Prokopets, and Uwe Aßmann. 2025. Context-Role Oriented Programming in Julia: Advancing Swarm Programming. In *IEEE/ACM 20th Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS*. 85–95. https://doi.org/10.1109/SEAMS66627.2025.00017

[35] Andrew M Hein, Sara Brin Rosenthal, George I Hagstrom, Andrew Berdahl, Colin J Torney, and Iain D Couzin. 2015. The evolution of distributed sensing and collective computation in animal populations. *eLife* 4 (2015), e10955. https://doi.org/10.7554/eLife.10955

[36] Francis Heylighen. 2016. Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research* 38 (2016), 4–13. https://doi.org/10.1016/j.cogsys.2015.12.002

[37] Mehmet F. Hocaoğlu. 2018. AdSiF: Agent driven simulation framework paradigm and ontological view. *Science of Computer Programming* 167 (2018), 70–90. https://doi.org/10.1016/j.scico.2018.07.004

[38] Luc Hogie. 2023. A Decentralized Web Service Infrastructure for the Interoperability of Applications in Multihop Dynamic Networks. In *6th Conference on Cloud and Internet of Things (CIoT)*. IEEE, 211–218. https://doi.org/10.1109/ciot57267.2023.10084876

[39] Xiangwang Hou, Jingjing Wang, Jun Du, Chunxiao Jiang, and Yong Ren. 2025. Distributed Machine Learning for Autonomous Agent Swarm: A Survey. *IEEE Commun. Surv. Tutorials* (2025), 1–1. https://doi.org/10.1109/comst.2025.3594713

[40] Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. 2007. Developing organised multiagent systems using the MOISE$^+$ model: programming issues at the system and agent levels. *Int. J. Agent Oriented Softw. Eng.* 1, 3/4 (2007), 370–395. https://doi.org/10.1504/IJAOSE.2007.016266

[41] Gerald Ijemaru, Li-Minn Ang, and Kah Phooi Seng. 2023. Swarm Intelligence Internet of Vehicles Approaches for Opportunistic Data Collection and Traffic Engineering in Smart City Waste Management. *Sensors* 23, 5 (2023), 2860. https://doi.org/10.3390/S23052860

[42] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. 2001. *Reactivity and Deliberation: A Survey on Multi-Robot Systems*. Springer, 9–32. https://doi.org/10.1007/3-540-44568-4_2

[43] Nick R. Jennings. 1993. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review* 8, 3 (1993), 223–250. https://doi.org/10.1017/s0269888900000205

[44] Iwens Gervásio Sene Júnior, Thalia S. Santana, Renato F. Bulcão-Neto, and Barry Porter. 2022. The state of the art of macroprogramming in IoT: An update. *J. Internet Serv. Appl.* 13, 1 (2022), 54–65. https://doi.org/10.5753/JISA.2022.2372

[45] Oussama Khatib. 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robotics Res.* 5, 1 (1986), 90–98. https://doi.org/10.1177/

027836498600500106
[46] Timothy A. Kohler, Darcy Bird, and David H. Wolpert. 2022. Social Scale and Collective Computation: Does Information Processing Limit Rate of Growth in Scale? *J. Soc. Comput.* 3, 1 (2022), 1–17. https://doi.org/10.23919/jsc.2021.0020
[47] Panagiotis Kouvaros and Alessio Lomuscio. 2013. Automatic Verification of Parameterised Interleaved Multi-Agent Systems. https://doi.org/10.48550/ARXIV.1301.6431
[48] Panagiotis Kouvaros and Alessio Lomuscio. 2015. Verifying Emergent Properties of Swarms. In *24th International Joint Conference on Artificial Intelligence, IJCAI*. AAAI Press, 1083–1089. http://ijcai.org/Abstract/15/157
[49] Christopher Landauer. 2025. Complexity Magnifiers Simplify Environment Models. In *IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, 94–99. https://doi.org/10.1109/acsos-c66519.2025.00035
[50] Jinsong Leng and Chee Peng Lim. 2011. Reinforcement learning of competitive and cooperative skills in soccer agents. *Appl. Soft Comput.* 11, 1 (2011), 1353–1362. https://doi.org/10.1016/J.ASOC.2010.04.007
[51] Dingbang Liu, Fenghui Ren, Jun Yan, Guoxin Su, Wen Gu, and Shohei Kato. 2024. Scaling Up Multi-Agent Reinforcement Learning: An Extensive Survey on Scalability Issues. *IEEE Access* 12 (2024), 94610–94631. https://doi.org/10.1109/ACCESS.2024.3410318
[52] Marco Mamei, Franco Zambonelli, and Letizia Leonardi. 2002. Co-Fields: Towards a Unifying Approach to the Engineering of Swarm Intelligent Systems. In *3rd International Workshop on Engineering Societies in the Agents World (LNCS, Vol. 2577)*. Springer, 68–81. https://doi.org/10.1007/3-540-39173-8_6
[53] Claudio Masolo, Laure Vieu, Roberta Ferrario, Stefano Borgo, and Daniele Porello. 2020. Pluralities, Collectives, and Composites. In *Formal Ontology in Information Systems (FOIS), 11th International Conference (Frontiers in Artificial Intelligence and Applications, Vol. 330)*. IOS Press, 186–200. https://doi.org/10.3233/FAIA200671
[54] Piero Mella. 2025. *The Theory of Combinatory Systems*. Springer Nature Switzerland, 1–81. https://doi.org/10.1007/978-3-031-86946-4_1
[55] Fabrizio Montesi. 2023. *Introduction to Choreographies*. Cambridge University Press. https://doi.org/10.1017/9781108981491
[56] Ryan Newton, Arvind, and Matt Welsh. 2005. Building up to macroprogramming: an intermediate language for sensor networks. In *4th International Symposium on Information Processing in Sensor Networks, IPSN*. IEEE, 37–44. https://doi.org/10.1109/IPSN.2005.1440891
[57] Rocco De Nicola, Stefan Jähnichen, and Martin Wirsing. 2020. Rigorous engineering of collective adaptive systems: special section. *Int. J. Softw. Tools Technol. Transf.* 22, 4 (2020), 389–397. https://doi.org/10.1007/s10009-020-00565-0
[58] Sascha Ossowski and Ronaldo Menezes. 2005. On coordination and its significance to distributed and multi-agent systems. *Concurrency and Computation: Practice and Experience* 18, 4 (2005), 359–370. https://doi.org/10.1002/cpe.943
[59] Alison R. Panisson, Rafael H. Bordini, and Antônio Carlos da Rocha Costa. 2020. A Multi-level Approach to the Formal Semantics of Agent Societies. In *Intelligent Systems - 9th Brazilian Conference, BRACIS (LNCS, Vol. 12320)*. Springer, 3–17. https://doi.org/10.1007/978-3-030-61380-8_1
[60] Shari Lawrence Pfleeger and Barbara Ann Kitchenham. 2025. Evidence-Based Software Engineering Guidelines Revisited. *IEEE Trans. Software Eng.* 51, 3 (2025), 814–819. https://doi.org/10.1109/TSE.2025.3526730
[61] Danilo Pianini, Martina Baiardi, Samuele Burattini, and Giovanni Ciatto. 2024. Multi-Paradigm Integration for the BDI Resurgence. In *IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, 27–28. https://doi.org/10.1109/acsos-c63493.2024.00025
[62] Ferdinand Piette, Costin Caval, Cédric Dinont, Amal El Fallah Seghrouchni, and Patrick Taillibert. 2016. *A Multi-agent Solution for the Deployment of Distributed Applications in Ambient Systems*. Springer International Publishing, 156–175. https://doi.org/10.1007/978-3-319-50983-9_9
[63] Carlo Pinciroli and Giovanni Beltrame. 2016. Buzz: A Programming Language for Robot Swarms. *IEEE Softw.* 33, 4 (2016), 97–100. https://doi.org/10.1109/MS.2016.95
[64] Yuansong Qiao, Robert Nolani, Saul Gill, Guiming Fang, and Brian Lee. 2018. ThingNet: A micro-service based IoT macro-programming platform over edges and cloud. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. 1–4. https://doi.org/10.1109/ICIN.2018.8401626
[65] Jette Randløv. 1998. Learning Macro-Actions in Reinforcement Learning. In *Advances in Neural Information Processing Systems 11, NIPS*. The MIT Press, 1045–1051. http://papers.nips.cc/paper/1586-learning-macro-actions-in-reinforcement-learning
[66] Fernando E. Rosas, Bernhard C. Geiger, Andrea I Luppi, Anil K. Seth, Daniel Polani, Michael Gastpar, and Pedro A. M. Mediano. 2024. Software in the natural world: A computational approach to hierarchical emergence. https://doi.org/10.48550/ARXIV.2402.09090
[67] Nick Russell, Wil M. P. van der Aalst, and Arthur H. M. ter Hofstede. 2016. *Workflow Patterns: The Definitive Guide*. MIT Press.
[68] R. Keith Sawyer. 2003. Artificial Societies: Multiagent Systems and the Micro-Macro Link in Sociological Theory. *Sociological Methods & Research* 31, 3 (2003), 325–363. https://doi.org/10.1177/0049124102239079
[69] Michael T. Schaub, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. 2019. Structured Networks and Coarse-Grained Descriptions: A Dynamical Perspective. , 333–361 pages. https://doi.org/10.1002/9781119483298.ch12
[70] Luca Serena, Moreno Marzolla, Gabriele D'Angelo, and Stefano Ferretti. 2023. A review of multilevel modeling and simulation for human mobility and behavior. *Simul. Model. Pract. Theory* 127 (2023), 102780. https://doi.org/10.1016/j.simpat.2023.102780
[71] David Servat, Edith Perrier, Jean-Pierre Treuil, and Alexis Drogoul. 1998. *When Agents Emerge from Agents: Introducing Multi-scale Viewpoints in Multi-agent Simulations*. Springer, 183–198. https://doi.org/10.1007/10692956_13
[72] Xiaoying Shi, Jiaming Zhang, Ziyi Liang, and Dewen Seng. 2023. MADDPGViz: a visual analytics approach to understand multi-agent deep reinforcement learning. *J. Vis.* 26, 5 (2023), 1189–1205. https://doi.org/10.1007/s12650-023-00928-0
[73] Armando Solar-Lezama. 2008. *Program synthesis by sketching*. University of California, Berkeley.
[74] Adrian Sosic, Wasiur R. KhudaBukhsh, Abdelhak M. Zoubir, and Heinz Koeppl. 2017. Inverse Reinforcement Learning in Swarm Systems. In *16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS*. ACM, 1413–1421. http://dl.acm.org/citation.cfm?id=3091320
[75] Aaron Hao Tan, Federico Pizarro Bejarano, Yuhan Zhu, Richard Ren, and Goldie Nejat. 2023. Deep Reinforcement Learning for Decentralized Multi-Robot Exploration With Macro Actions. *IEEE Robotics Autom. Lett.* 8, 1 (2023), 272–279. https://doi.org/10.1109/LRA.2022.3224667
[76] Vito Trianni. 2008. *Evolutionary Swarm Robotics - Evolving Self-Organising Behaviours in Groups of Autonomous Robots*. Studies in Computational Intelligence, Vol. 108. Springer.
[77] Samuel H. Christie V., Munindar P. Singh, and Amit K. Chopra. 2023. Kiko: Programming Agents to Enact Interaction Models. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. ACM, 1154–1163. https://doi.org/10.5555/3545946.3598758
[78] Gabriele Valentini, Heiko Hamann, and Marco Dorigo. 2022. Global-to-Local Design for Self-Organized Task Allocation in Swarms. *Intelligent Computing* 2022 (2022). https://doi.org/10.34133/2022/9761694
[79] Mirko Viroli, Jacob Beal, Ferruccio Damiani, Giorgio Audrito, Roberto Casadei, and Danilo Pianini. 2019. From distributed coordination to field calculus and aggregate computing. *J. Log. Algebraic Methods Program.* 109 (2019). https://doi.org/10.1016/J.JLAMP.2019.100486
[80] Terry Winograd. 1979. Beyond programming languages. *Commun. ACM* 22, 7 (1979), 391–401. https://doi.org/10.1145/359131.359133
[81] Tom De Wolf and Tom Holvoet. 2007. Designing Self-Organising Emergent Systems based on Information Flows and Feedback-loops. In *1st International Conference on Self-Adaptive and Self-Organizing Systems, SASO*. IEEE Computer Society, 295–298. https://doi.org/10.1109/SASO.2007.16
[82] David H. Wolpert and Kagan Tumer. 1999. An Introduction to Collective Intelligence. arXiv:cs/9908014 [cs.LG] https://arxiv.org/abs/cs/9908014
[83] Michael Wooldridge. 2009. *An introduction to multiagent systems*. John wiley & sons.
[84] Daniel Yamins and Radhika Nagpal. 2008. Automated global-to-local programming in 1-D spatial multi-agent systems. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Vol. 2*. IFAAMAS, 615–622. https://dl.acm.org/citation.cfm?id=1402310
[85] Ruilin Yao, Bo Zhang, Jirui Huang, Xinwei Long, Yifang Zhang, Tianyu Zou, Yufei Wu, Shichao Su, Yifan Xu, Wenxi Zeng, Zhaoyu Yang, Guoyou Li, Shilan Zhang, Zichan Li, Yaxiong Chen, Shengwu Xiong, Peng Xu, Jiajun Zhang, Bowen Zhou, David Clifton, and Luc Van Gool. 2025. LENS: Multi-level Evaluation of Multimodal Reasoning with Large Language Models. https://doi.org/10.48550/ARXIV.2505.15616
[86] Bing Yuan, Jiang Zhang, Aobo Lyu, Jiayun Wu, Zhipeng Wang, Mingzhe Yang, Kaiwei Liu, Muyun Mou, and Peng Cui. 2024. Emergence and Causality in Complex Systems: A Survey of Causal Emergence and Related Quantitative Studies. *Entropy* 26, 2 (2024), 108. https://doi.org/10.3390/e26020108
[87] X.F Zha. 2002. A knowledge intensive multi-agent framework for cooperative/collaborative design modeling and decision support of assemblies. *Knowledge-Based Systems* 15, 8 (2002), 493–506. https://doi.org/10.1016/s0950-7051(02)00034-5
[88] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. Springer International Publishing, 321–384. https://doi.org/10.1007/978-3-030-60990-0_12
[89] Sai Zhang, Zhenchang Xing, Ronghui Guo, Fangzhou Xu, Lei Chen, Zhaoyuan Zhang, Xiaowang Zhang, Zhiyong Feng, and Zhiqiang Zhuang. 2025. Empowering Agile-Based Generative Software Development through Human-AI Teamwork. *ACM Trans. Softw. Eng. Methodol.* 34, 6 (2025), 156:1–156:46. https://doi.org/10.1145/3702987
[90] Yutian Zhang, Guohong Zheng, Zhiyuan Liu, Quan Li, and Haipeng Zeng. 2025. MARLens: Understanding Multi-Agent Reinforcement Learning for Traffic Signal Control via Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 31, 7 (2025), 4018–4033. https://doi.org/10.1109/tvcg.2024.3392587