Applied Intelligence (2018) 48:1251–1274
https://doi.org/10.1007/s10489-017-1112-z

Agent systems verification : systematic literature review and mapping

Najwa Abu Bakar[1] · Ali Selamat[1,2,3]

Published online: 13 January 2018

Abstract

Agent systems are distributed systems consist of agents that autonomously interact to each other in an environment to perform tasks and achieve goals. Performing verification is important to ensure correctness of agent properties and to detect faults. The objective of this review is to identify research gap and future research direction of agent systems verification. In this study, the surveys of existing techniques for checking agent properties and detecting faults during design, development and runtime phases of agent system life-cycle are presented. Search terms with relevant keywords were used to identify primary studies that relate to the topic of discussion. Next, the studies were classified based on the used techniques and the addressed properties. 231 primary studies were identified during the search process. From these studies, 49% were implemented for verification of agent systems during design, 27% during development and 25% during runtime. Model checking or model-based verification techniques are the highest proposed techniques (44%) followed by the testing and debugging during development (17%). The properties that are largely addressed by the selected studies are temporal properties (19%) and epistemic properties (9%). At the end of the review, the research gap and the future research direction are presented.

Keywords Agent systems · Multi-agent systems · Verification · Testing · Debugging · Fault management · Violations detection

1 Introduction

Agent systems are inherently complex distributed systems consist of agents that autonomously interact to each other in an environment to perform tasks and achieve goals [151]. Performing verification is important to detect violations of agents and the agent systems properties. Verification of agent systems is the process to check properties violations of agents complex systems. In general, the verification is performed by mapping the requirements of the agent systems into properties specifications that will be checked using verification techniques whether real system or system design model comply with the specified properties. The systems that meet or fulfill the specifications are classified as correct systems [41].

This paper reports the completed Systematic Literature Review (SLR) that explores the system verification techniques proposed so far and classifies as well as analyzes them. The objective of this review is to identify gap and future research direction in the area of agent systems verification. This study aims to identify existing verification techniques and the properties addressed by those techniques. The surveys of existing techniques for checking agents and the agent systems properties during design, development and runtime of agents and the agent systems lifecycle are presented. This paper also identifies what are the criteria for each agent systems property in which the checking rules (to decide properties violations) are based on. Search terms with relevant keywords were used to identify primary studies that relate to the topic of discussion classified based on the used techniques and the addressed properties.

Existing traditional literature reviews of agent systems verification research are mainly focusing on specific techniques and properties. In our study, we have performed

✉ Ali Selamat
aselamat@utm.my

Najwa Abu Bakar
najwa.abakar@gmail.com

1 Faculty of Computing, Universiti Teknologi Malaysia, 81310, Johor Bahru, Johor, Malaysia

2 Malaysia & Media and Game Innovation Center of Excellence (MaGICX), Universiti Teknologi Malaysia, 81310, Johor Bahru, Johor, Malaysia

3 FIM, Center for Basic and Applied Research, University of Hradec Kralove, Rokitanskeho, 62, Hradec Kralove, 500 03, Czech Republic

an SLR to address sets of research questions in order to extract related information from the selected literatures. This SLR is intended to complement the traditional approach as its advantage is to give thorough state-of-the-art overview as compared to traditional literature review that provides an in-depth study, which is more specialized only toward certain techniques or properties. We performed an SLR on verification of agent systems published from 1995 to January 2017. In summary, the aim of this SLR is to provide wide overview and summary of (1) the existing agent systems verification techniques, (2) the agent systems life-cycle phases where the techniques have been implemented, and (3) the focused agents and agent systems properties. This SLR is intended to benefit agent systems designers, developers, and researchers from the industries and academia to understand the research pattern of agent systems verification practices and overcome the challenges.

The rest of the paper is structured as follows. Section 2 presents the existing reviews. Section 3 describes the SLR methodology. Next, Section 4 presents the SLR results, maps the surveyed techniques with the identified properties and discusses the SLR results. Finally, Section 5 explains the SLR findings, and Section 6 concludes the SLR.

2 Background and related works

System verification ensures that the system is developed according to user requirements and design specifications (building the system right). Typical systems verification techniques are formal verification and runtime verification. Formal verification methods use mathematical proofs to perform verification. There are two types of formal verification that are deductive reasoning and model checking. Deductive reasoning such as manual proof and theorem proving proves the correctness of a system using axioms and proof rules while model checking checks correctness of system with respect to its requirement specifications [185]. To perform verification using model checking, the system is modeled as a state transition graph while the requirements in the form of properties specifications are expressed in logic notation. The model of a system is checked automatically whether it satisfies the expressed properties [62].

The existing approaches to verify agents and agent systems properties have been reviewed in several works. Janssen [110] explored and discussed the general criteria to evaluate multi-agent systems models verification and the challenges encountered. Bordini et al. [41] discussed current issues in multi-agent systems development including verification using model checking and stressed out the importance of having verification tools that are specific for verifying multi-agent systems applications. a Al [1] compared the existing verification techniques for multi-agent systems that are formal, semi-formal, hybrid and conventional techniques. The paper also stated how the verification of multi-agent systems properties process should be executed and what the expected outcomes were.

Many existing reviews of multi-agent systems verification techniques have been focusing on testing approach. Góxmez-Sanz et al. [96] explained the testing techniques (via verification and debugging) of multi-agent systems that are parts of multi-agent systems development process to identify failures and to check whether the system satisfies requirements. Athanázio et al. [89] surveyed the testing and debugging approaches and evaluating them based on the strategies used such as black-box, white-box, progressive, regressive, and performance testing. Multi-level testing that are at agent level (agents functionality) and society level (cooperation and coordination) were also reviewed. Houhamdi [105] has performed a survey to analyze the existing approaches and challenges of multi-agent systems testing. Nguyen et al. [181] classified multi-agent systems testing works based on the supported testing levels that are unit, agent, integration, system, and acceptance levels. The techniques used were also classified into passive (simulation-based techniques) and active (structured testing) approaches [181]. Zamani [233] investigated the important criteria of multi-agent systems testing and performed an evaluation of existing testing methods. The paper has identified the testing criteria that were disregarded by most of the multi-agent systems testing methods in order to propose standard testing process. Moreno et al. [168] and [1] gave the overview of verifications for multi-agent systems properties as continuous activities such as unit test, module test, integration test, system or functional test, and acceptance test throughout the entire multi-agent systems life-cycle that include the requirement level, design level, and implementation level. Both works presented the multi-agent systems testing process based on V-Model [112] to classify relevant verification methods and tools as well as identifying multi-agent systems characteristics that have not been tested [168].

There are also reviews and classification of faults and properties violation in agent systems that could be detected during runtime verification. Potiron et al. [195] extends the existing faults classification (development faults, physical faults and interaction faults) by adding specific faults related to agents internal and external autonomous behavior. These faults should be considered by developers for specification in order for the faults to be detected and to implement fault tolerance [195]. Bijani and Robertson [35] performed review of the existing practices to detect confidentiality, integrity, availability, and accountability violations in open multi-agent systems.

The above mentioned works have performed reviews of existing approaches and techniques for verifying agent systems properties and the classification of faults and properties violations in agent systems. Majority of the reviews have been focusing on the testing techniques [89, 96, 105, 168, 181, 233] and only two surveys classified multi-agent systems faults and properties violations for detection [35, 195]. However, the existing verification works of multi-agent systems have been widely performed in every stage of agent systems life-cycle and implemented for multiple types of properties using various kinds of techniques and tools that need thorough surveys and further analysis. Therefore, there is a need for a more comprehensive and systematic literature review to see the pattern of publications in this area to recognize, classify, and map the verification techniques and agent systems properties addressed by researchers to date.

3 Systematic Literature Review (SLR) methodology

Systematic Literature Review (SLR) is defined by [122] as “a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest” . Basically, the main purpose of performing the SLR is to study the width and the depth of the researched topic. In other words, SLR is performed to identify the related properties of certain topic and to reveal what are the issues and problems in each element. Systematic mapping has been also introduced as part of SLR approach [122]. It is used to construct a visual summary or map of a structure of the types of literatures and the published results by categories. The mapping process has been previously performed by [4, 158] to classify literatures according to types of papers and published results. In this study, the mapping approach is used to identify the properties of multi-agent systems interaction and to categorize the existing works focusing on verifying those properties. The process flow diagram of the systematic mapping is shown in Fig. 1 and the detail of the process is explained in the following subsections.

3.1 Definition of review questions (Scope)

To perform research in this area and to propose a new solution, the existing literatures need to systematically reviewed to identify gap. Researches in the area of agent system verification comprised of three main contributions that are i) the objectives (outcomes) of the verification, ii) the verification techniques used, and iii) the agent or agent system properties being verified. The SLR research questions are defined with the purpose to extract the information of the above three contributions from the selected literatures. The information will be used to perform mapping and to construct visual summary that shows the relation between the three contributions mentioned above. Therefore, the research questions (scope) addressed in this review are:

RQ1: What are the verification techniques used by each work?
RQ2: During which life-cycle phases the techniques were intended to be used?
RQ3: What are the objectives of the verification and what are the properties focused by each work?
RQ4: What are the outcomes of the data extractions and classifications of the verification works?

3.2 Conduct searching for primary studies in the literatures

The search strategy utilized in this research consists of the process of deriving the search terms and executing the search process at the selected literature resources. Similar to the research questions, this searching process also should extract information related to verification objectives, used techniques, and focused properties in the area of agent systems. Agent systems are also referred as multi-agent systems in many literatures to reflect that there are many agents in a system. The process to derive the search strings includes:

(a) Identifying the primary terms that are: “agent system” and “multi-agent system”.
(b) Identifying the terms that are related to research questions. In this study we include the techniques that are normally used for verification in the search keywords that are: “verification”, “testing”, “debugging”, “monitoring”, “assessing”, “diagnosing”, and “detecting”.
(c) Identifying the terms used for describing the research outcomes. The word “violations”, “correctness”, “faults”, “failure”, “error”, “anomaly”, “emergent”, “invalid”, and “unwanted” that are the outcomes for verification process are included.
(d) Using of the Boolean OR to incorporate alternative technique names.
(e) Using of the Boolean AND to link the major terms.

The resulting search terms are described as follows: (agent system OR multi-agent system) AND (verification OR testing OR debugging OR monitoring OR assessing OR diagnosing OR detecting) AND (violations OR correctness OR faults OR failure OR error OR anomaly OR emergent OR invalid OR unwanted). The manual search steps performed are presented below:

Step1: Keyword search of the related journals such as Journal of Autonomous Agents and Multi-Agent Systems and Journal of Artificial Intelligence Research, SCOPUS, IEEE/IET Electronic Library

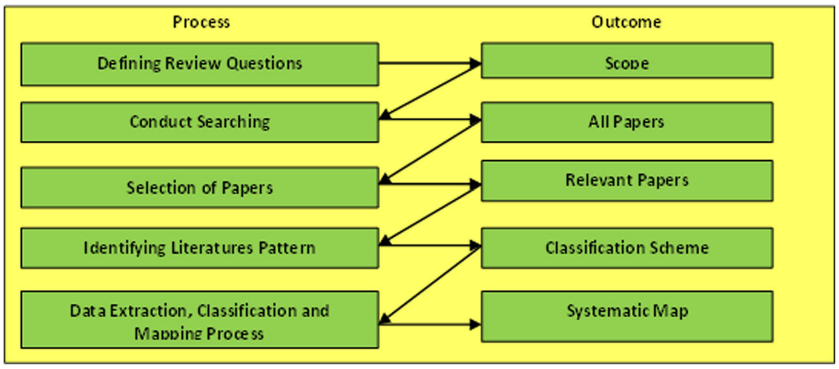

Fig. 1 Process flow of the systematic mapping

(IEL), and ACM Digital Library (ACM DL), Inspec bibliographic databases using Serial Solutions Central Search.

Step2: Google Scholar search using the same search strings with filter to search only in Engineering, Computer Science, and Mathematics

3.3 Selection of papers and searching additional papers (Relevant Papers)

The inclusion criteria covered the studies related to the techniques used for verification of agents and agent systems properties. The studies of verification that are not related to the verification of agents and agent systems properties are excluded. The inclusion and exclusion criteria as the basis to include and exclude the studies are listed below:

Inclusion:

(i) Papers that match the search keywords and addressing the research questions
(ii) Papers that explicitly propose techniques for verifying agents and agent systems properties
(iii) Papers that explicitly focus on verifying of agents and agent systems properties

Exclusion:

(i) Papers such as surveys and reviews that match the search keywords but only give general overview and do not focus on proposing solutions for agent systems verification
(ii) Papers that focus on verification of properties of systems that are not agent-based or multi-agent systems.

3.4 Identifying the research pattern (Classification Scheme)

As this SLR aims to analyze the proposed solutions for agents and agent systems properties verifications, several classification steps have been identified. The information are classified based on the reported techniques, properties, life-cycle levels where the techniques would be implemented, and the verification goals either for checking properties satisfaction (correctness) or detecting properties violations (faults). This classification approach was chosen for the following purposes:

(i) To analyze the existing works in the area of agent systems verification in order to identify the needs to extend the existing approaches and propose new contribution.
(ii) To accomplish the objective that is to find gap and derive research direction.
(iii) To report the number of existing works that have been focusing on certain techniques, properties and outcomes.
(iv) To map, visually summarize and to identify which techniques have been largely used, for which properties and for what purposes.

Other alternative classification approaches may also be used such as classifications of the literatures based on the types of publications (journals, conferences, etc) or the types of works reported (methods, models, frameworks, case studies, etc). However, to analyze previous works for finding research gap, deriving research direction and proposing new contribution toward new techniques and identify new properties to be verified, the classifications need to be made according to the existing reported techniques, properties, life-cycle levels, and outcomes. Based on those reported works, the classifications need to be performed based on the patterns found when the data has been extracted from the selected literatures.

First, the selected papers are classified according to the types of verification techniques used for checking agents and agent systems properties. Second, the techniques are further classified into the agent systems life-cycle levels where the techniques were intended to be used. Third, the papers are classified based on agents and agent systems properties addressed in the papers. Fourth, the properties are grouped based on the verification approach or goals that are either to check properties satisfactions or to detect properties violations. Finally, the properties addressed in the papers were classified into agents and agent systems properties groups that are (i) system properties, (ii) agents internal properties, (iii) agents social properties, (iv) agents knowledge properties, and (v) agents interaction properties as detailed out below:

(i) System properties include the overall correctness of the agent systems such as temporal properties, reachability, events sequence and concurrent properties.
(ii) Agent internal properties include the behavior of agents actions, characteristics and capabilities as individual agents such as reactiveness, proactiveness and controllability.
(iii) Agent social properties include agents social behavior among group of agents such as commitment, coordination and self-organizing.
(iv) Agent knowledge properties include the properties related to agents cognitive or mental states such as belief, goal and epistemic properties.
(v) Agent interaction properties include the correctness of the agents interaction protocol, language, message exchange, message sequence, message syntax and message content.
(vi) Agent quality properties include non-functional requirements and constraints to protect and to ensure that the agents can be trusted such as security. safety, privacy, confidentiality and integrity.

3.5 Data extraction and mapping process (Systematic Map)

Data extraction is a stage performed in systematic reviews. It is the process of gathering relevant data from the selected resources. Figure 2 illustrates the data extraction, classification and mapping process flow. For this study, the data extracted include the techniques, properties, lifecycle levels and verification goal (outcomes). In order to perform the classification steps mentioned in the previous section, the information from the papers need to be extracted. The information are classified based on the techniques, the properties, the life-cycle levels where the techniques would be implemented, and the verification goals either for checking properties satisfaction (correctness) or detecting properties violations (faults). Finally, the properties groups and the techniques types are then mapped to summarize and see which techniques have been largely used and for which properties.

4 Systematic literature review results

This section presents and discusses the findings of this review. Firstly, an overview of the selected studies are presented. Secondly, a detailed description of the findings of this review in line with the specified research questions are presented in separate sub-sections. Finally, the review results are also interpreted in this section, in the context of the research questions.

4.1 Overview of the searched and selected studies

Figure 3 shows the searching and selection process. During the first searching process, 358 papers were collected. After removing duplicates and applying the inclusion and exclusion criteria, the total of relevant papers were 193. Next, the second searching process of the papers cited by the previously selected papers was performed and 231 papers were collected.

Figure 4 presents the number of works by year of publication. For the past two and half decades, lots of researches have been performed in the area of agent and agent system verification. From 2004 until January 2017, the number of publications are very high (ranging between

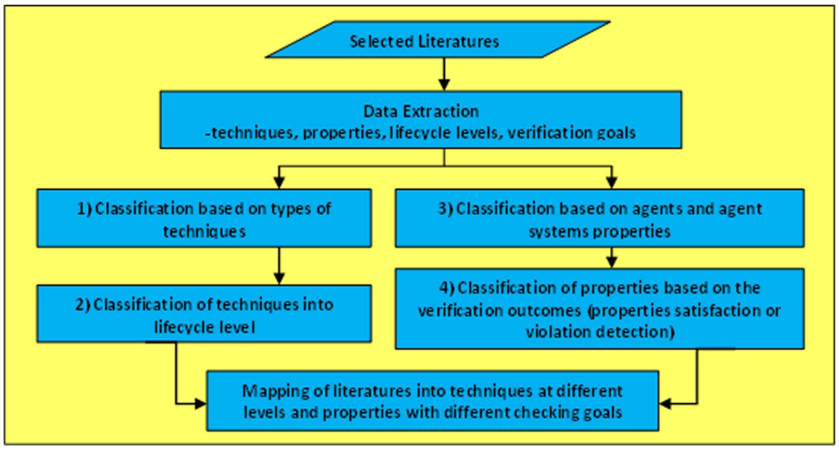
Selected Literatures
Data Extraction
-techniques, properties, lifecycle levels, verification goals
1) Classification based on types of techniques
3) Classification based on agents and agent systems properties
2) Classification of techniques into lifecycle level
4) Classification of properties based on the verification outcomes (properties satisfaction or violation detection)
Mapping of literatures into techniques at different levels and properties with different checking goals

Fig. 2 Data extraction, classification and mapping process flow

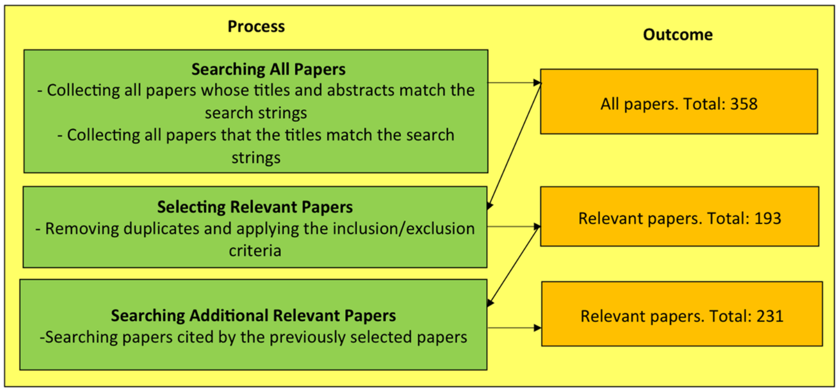
Process
Outcome
Searching All Papers
- Collecting all papers whose titles and abstracts match the search strings
- Collecting all papers that the titles match the search strings
Selecting Relevant Papers
- Removing duplicates and applying the inclusion/exclusion criteria
Searching Additional Relevant Papers
-Searching papers cited by the previously selected papers
All papers. Total: 358
Relevant papers. Total: 193
Relevant papers. Total: 231

Fig. 3 Searching and selection process

10 to above 25 papers) every year in which 2015 shows the highest number of publications. At this time while the paper is written, more publications are being published in the area of agent and agent system verification. The number of publications shows that agent system verification area is an active research area worth to be explored and studied.

The SLR processes include the identification of agent properties focused by researches, the identification of verification techniques used and the classifications of the techniques and the properties. From the classifications, research gap in the area of agent systems verification is identified and research direction is derived at the end of the SLR. The results of the process that answer each of the research questions are presented in the following subsections.

4.2 Identified verification techniques (RQ1)

From the selected 231 relevant papers, the techniques proposed and used by each paper were identified. Figure 5 shows number of works for each type of techniques used for checking agent properties. The techniques were grouped into 10 main types of verification techniques for easy analysis. The most popular technique is the model checking, formal method, or model-based verification technique as 44.04% (96) of the works use this type of techniques to verify agent and agent system properties. This is followed by the testing and debugging agent and agent system properties during development as 16.51% (36) of the works address this type of techniques to check agent and agent system properties. The next popular technique is the runtime verification as 9.17% (20) of the works choose this type of techniques.

Existing researches have achieved their objectives in addressing agent and agent system properties. Table below shows the existing achievements and limitations of verification techniques in addressing agent properties [38].

4.3 Classification of the techniques according to life-cycle levels (RQ2)

The techniques for verifying agent and agent system properties were proposed to be implemented during certain multi-agent systems and agent systems life-cycle phases. Three agent systems life-cycle levels were identified that are design level, development level and runtime level where the

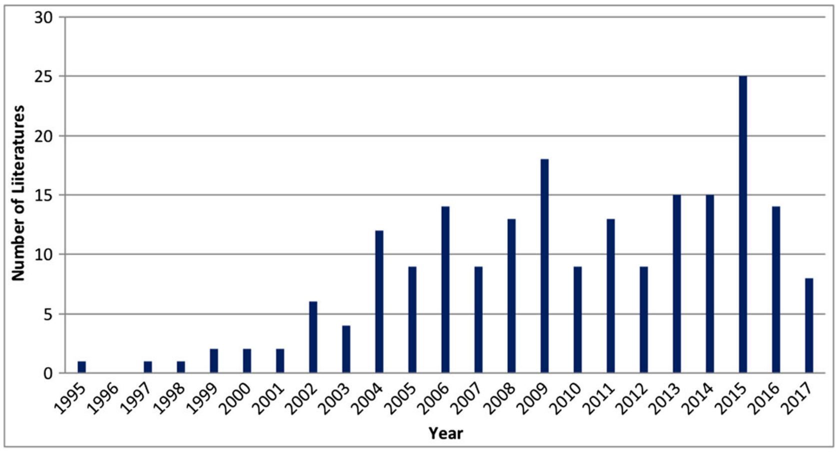

Fig. 4 Number of works by year of publications (1995-2017)

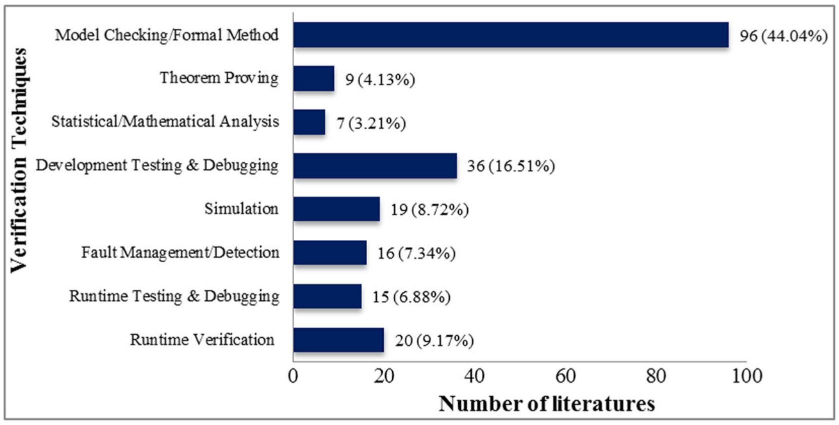

Fig. 5 Number of works proposing verification techniques

techniques are intended to be used as suggested by [24]. The techniques were classified into these three agent systems life-cycle levels. Percentages of the existing agent and agent system verification works implemented in those three levels are displayed in Fig. 6. Most of the verification activities were performed during design level (49%). It is followed by techniques that had been proposed to be implemented during development level (27%) and design level (25%).

The techniques utilized in each level were identified. First, the techniques that were implemented during design are model checking, model-based verification, formal method, theorem proving, mathematical analysis and statistical analysis. Second, during development, testing, debugging and simulation techniques are used. Third, during runtime, model-based fault detection, fault/anomaly management, testing, debugging, runtime verification, execution checking and analysis using logical, statistical, empirical and artificial intelligence (AI) techniques are utilized. Figure 7 shows the techniques that have been implemented during each life-cycle phase.

Based on the above classification, the issues of agents and agent systems can be divided into three phases that are during design, during development and during runtime. During design, it is possible that there are design flaws such as deadlock, live-lock, underspecification, over-specification, and interaction protocol violations [2]. Since 1990s, many solutions have been proposed. One of them is by inventing tools e.g. MCMAS (model checker) [147] and SOCS-SI (automated theorem proving) [77] customized to perform formal verification on agents and agent systems during design time by considering common agent abstractions [24]. Another approach is by checking on selected critical smaller models extracted from the complex agents and agent systems and use general-purpose distributed model checking tools such as SPIN [104], UPPAAL [219], NuSMV [61] and PRISM [131] to verify against general properties such as temporal logic. Many research works have been done on agent and agent system verification during design using model checking. Benerecetti and Cimatti [32] have proposed the use of MATL (Multi-Agent Temporal Logic) and MAFSM (Multi-Agent Finite State Machine) for the specification of multi agent BDI (beliefs, desires, and intentions) properties and modeling of the multi agent system, respectively. Next, series of works performed by [40] extend the temporal specifications of agent properties with the agent behavior properties of BDI. The specifications are verified using Spin and later using Agent JPF (AJPF) model checker. Another

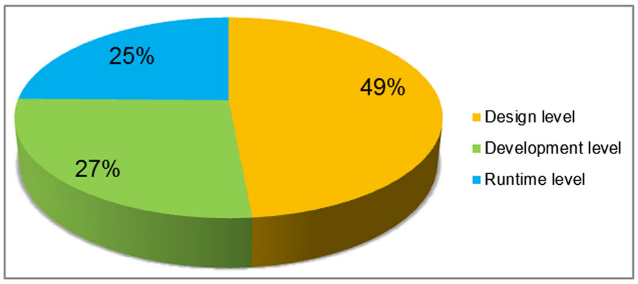

Fig. 6 Agent system verification implementation

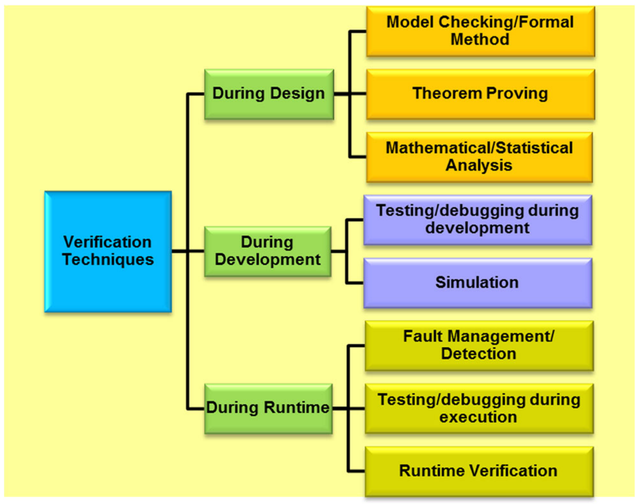
Verification Techniques
During Design
During Development
During Runtime
Model Checking/Formal Method
Theorem Proving
Mathematical/Statistical Analysis
Testing/debugging during development
Simulation
Fault Management/ Detection
Testing/debugging during execution
Runtime Verification

Fig. 7 The techniques implemented at different lifecycle levels

work that supports the verification of BDI properties for multi-agent systems is MABLE [223], a multi agent programming language and model checking. Lomuscio et al. [147] developed model checker for multi-agent systems (MCMAS) that can verify time, knowledge, and behavior of agents. Similarly, MCK (Model Checking Knowledge) [87] is also developed to reason about temporal (time) and epistemic (knowledge) properties.

During development, developed agent systems need proper debugging and testing. Monitoring or debugging of agents and agent systems has been performed for individual and multi-agent levels. Individual agent testing and debugging can be accomplished using debugging and testing techniques, platforms or tools for the programming languages used to develop the agents such as JAVA using JADE platform [31] or AgentSpeak using Jason [38]. Multiple agents need debugging and testing that includes the preparation of test cases of agent behavior, knowledge, and interaction during development phase. For fewer number of agents (less than 100), interactions between agents can be monitored using existing agent platforms such as JADE. On the other hand, for higher number of agents, advanced high-level analysis such as data mining is performed toward the ACL messages using tools such as ACLAnalyser [45]. During runtime, agents interact with each other by sending request and response messages, distributing and publishing information to participate, contribute, or collaborate in agent community. Agent communication can be specified to follow multi agent standard interaction protocols specified by FIPA. In addition, to understand each other at the application level, a common ontology specifying languages and vocabulary used within the application has to be followed. As the system operations, environment, and user inputs evolve, there is a pressing need for verification during runtime [23].

During runtime, executed system can still suffer from properties violations during runtime due to external factors such as dynamic input and preferences from users, intruders, newly integrated software, or other agents from different platforms, hosts, networks, or environments (web services, Internet or wireless network). The issues include correctness of message structure (syntax), correctness of message content (ontologies), correctness of message sequence or conversation (interaction protocol), believability (trust toward the sender), confidentiality (role of the receiver), etc. Thus, besides verification during design, monitoring and verification during runtime are equally important to verify the correctness of agent communication. Interactions between agents from different hosts and platforms can be visualized using JADE RMA (Remote Monitoring Agent) graphical user interface (GUI). The produced sequence diagrams allow message sequence correctness to be checked and analyzed to certain level [31]. As agent system or multi-agent systems itself is a solution for critical problems for various kinds of social, business, and computer applications [215]. The verification of agent systems is exceptional from other systems [41]. Model checking agents can always be developed within the applications, alongside other agents in multi-agent systems to verify running processes. Model checking method called MCa (Model Checking Agent) [206] and MVA (Model Verification Agent) [207] have been developed to verify message passing properties such as byte, syntax, and time checking of SMS Management and RFID systems, respectively. Osman [185] introduced runtime verification for agent interactions that check agent deontic and trust models. Finally, [198] proposed a debugging process structure and the implementation of debugging agent to detect commonly found errors during multi-agent systems execution such as uninitialized agent, failure to send messages, wrong recipients, message sent multiple times, and wrong message sent. The proposed debugging agent monitors exchanged agent interaction messages and checks their correctness against interaction protocols [198].

4.4 Identified verification objectives and the focused agent properties (RQ3)

From the searched and selected relevant agent system verification literatures, the verification objectives for each paper are identified. The aims of the proposed verification solutions are classified into two main objectives that are 1) to check satisfaction of agent properties, and 2) to detect agent properties violations (faults). Based on these two objectives, the identified properties are classified into two large groups that are 1) the checked agent properties and 2) the detected properties violations (faults). Figure 8 shows the number of works that check satisfaction of the agent properties. The most popular properties are the temporal properties as 47 research works had been focusing to verify these properties. Next, 23 works for verifying epistemic properties had been performed, followed by agent safety (15 works), interaction protocol (12 works), liveness (12 works) and agent BDI properties (10 works). Other properties such as reachability (8 works) and security (8 works) are also significant in the area of agent systems verification.

Beside satisfaction or correctness checking of the properties, the verification activities also focused on detecting faults as shown in Fig. 9. Most faults detected are the interaction pattern and message sequence faults (14 works). Faults that are due to emergent or unexpected behaviors (13 works) and message exchange (10 works) are the next popular issues being focused by researchers. Belief-related faults (8 works), goal-related faults (7 works) and interaction protocol violations (6 works) also contributed to the number of faults detected during agent systems

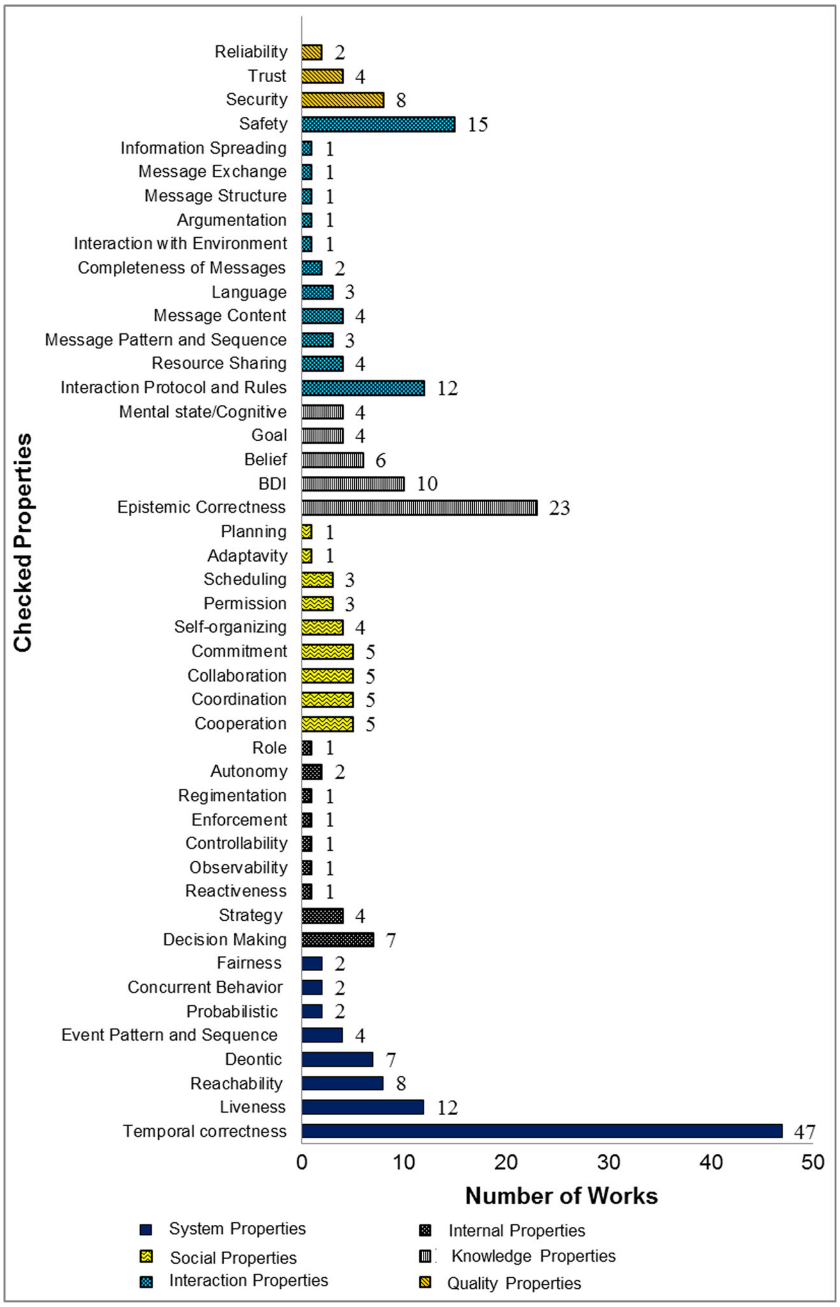

Fig. 8 Number of works that check satisfaction of agent properties

verification. The faults are also classified into 5 main groups that are agent system faults, agent internal faults, agent social faults, agent knowledge faults, and agent interaction faults. However, agent quality faults have yet to be detected.

The classification of the properties extracted from the literatures is shown in Fig. 10. The properties are classified into six main properties groups that are the agent system properties, internal properties, social properties, knowledge properties, interaction properties, and quality properties. The classification was made based on the words mentioned or used in existing papers. This classification was intended to see the popularity of the properties addressed by existing works.

Next, the agent properties are further classified into agent requirement categories that are agent functional and non-functional requirements [49, 232]. Figure 11 shows the agent properties classified under agent functional and non-functional properties. The agent functional properties are the properties that fulfill agent functional requirements and agent non-functional properties are the properties that should satisfy agent non-functional (quality) requirements.

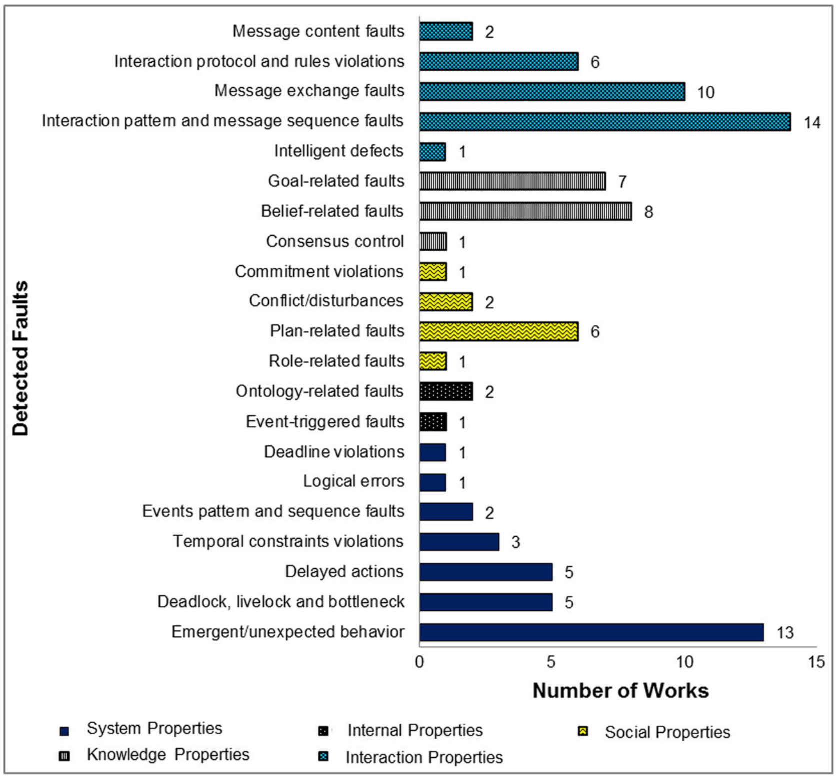

Fig. 9 Number of works that detect properties violations (faults)

4.5 The mapping of the verification works classifications (RQ4)

The SLR results have presented the classification of verification techniques into several groups (RQ1), the classification of works into lifecycle where the techniques had been used (RQ2), and the classification of works into verification objectives and into categories of focused properties (RQ3). For RQ4, the outcomes of the approaches are classified into checked properties satisfactions and detected properties violations. The results of the categorization of the papers into verification outcomes and focused properties are included in Appendix A. Consequently, a map of verification techniques is developed with respect to the checked properties and detected properties violations (faults) as shown in Fig. 12. The map represents the visual summary of the mostly used techniques and the successfully checked properties and detected faults.

The results of the mapping show distribution of the number of works using certain techniques for checking agent properties or detecting faults. Large number of works had been dedicated to check properties during design (using model checking or model-based verification, theorem proving, mathematical analysis, and formal method) as compared to efforts to verify properties during development (using simulation, debugging, and testing) and runtime (fault or properties violations detection, fault management, debugging, testing, and runtime verification). It is obvious that the model checking or model-based verification techniques and formal methods have been the main focus of the researchers in this area as majority of the works had been addressing these techniques for the checking of agent properties (142 works). The debugging or testing techniques performed during development proposed to detect faults and properties violations of agent properties (40 works) are the second highest number of classified works. The third group is the runtime verification techniques used to check satisfactions of agent properties (21 works each). The mapping shows that the majority of the verification works are focusing on the checking of properties while fewer efforts are directed to detect faults or properties violations in agents and agent systems. The map also shows that there is an implementation gap for the purpose of detecting faults of agent quality properties.

5 SLR findings and research direction

Verification of agent and agent system properties is a huge topic as improvement of techniques are being proposed and new properties are being discovered form time to time. 231 primary studies were identified during the search process. From these studies, 49% were implemented for verification of agents and agent systems during design, 27%

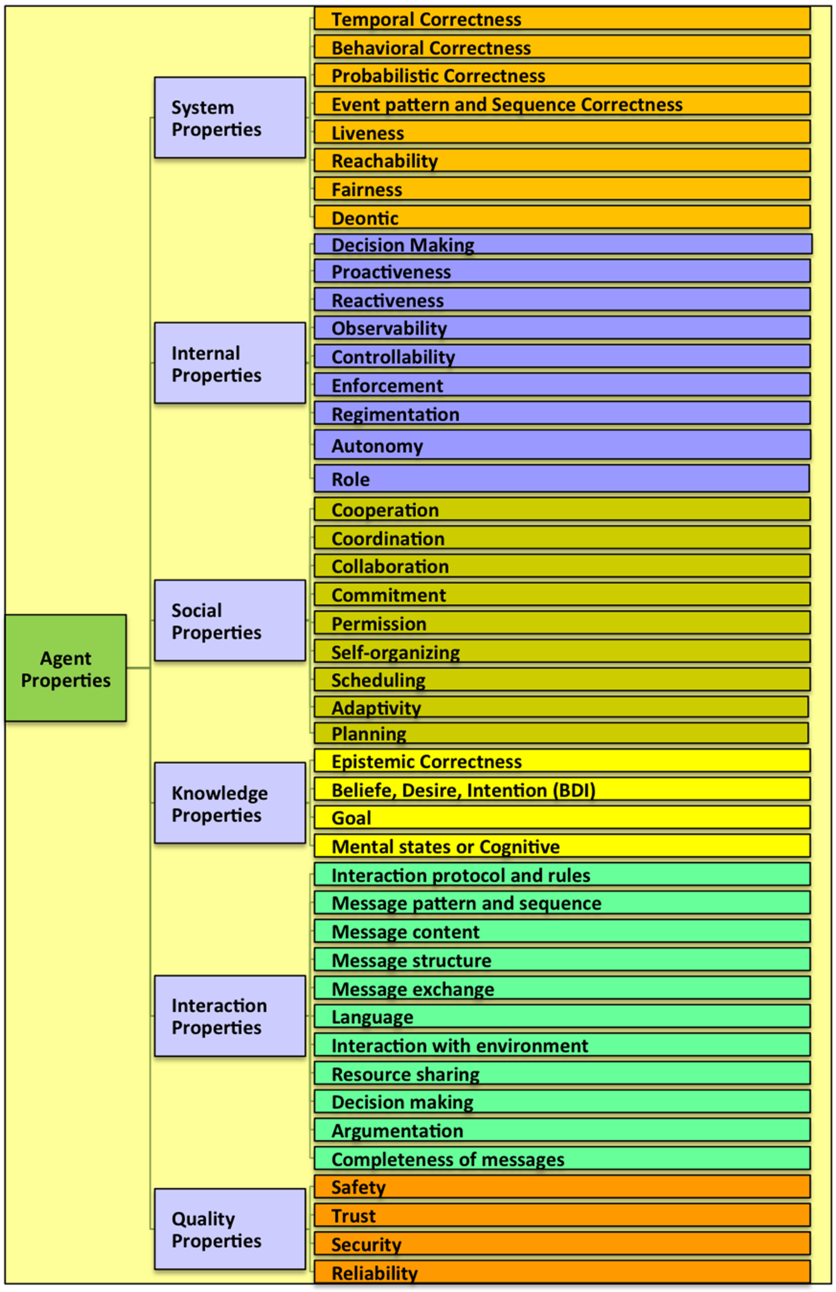

Fig. 10 Properties checked by existing works

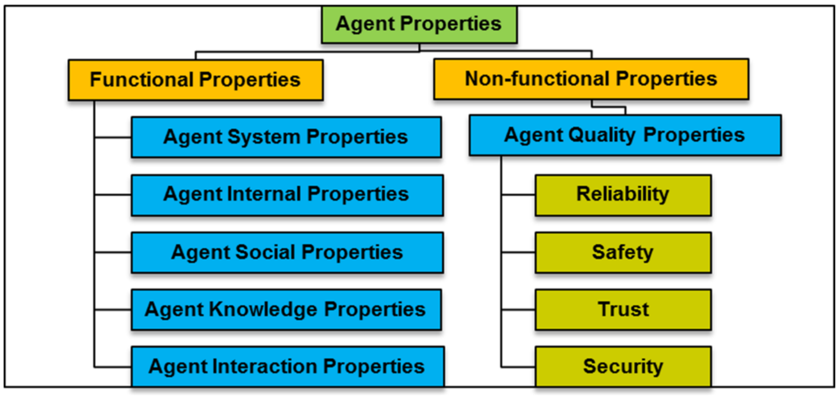

Fig. 11 Classification of agent properties based on agent requirement categories

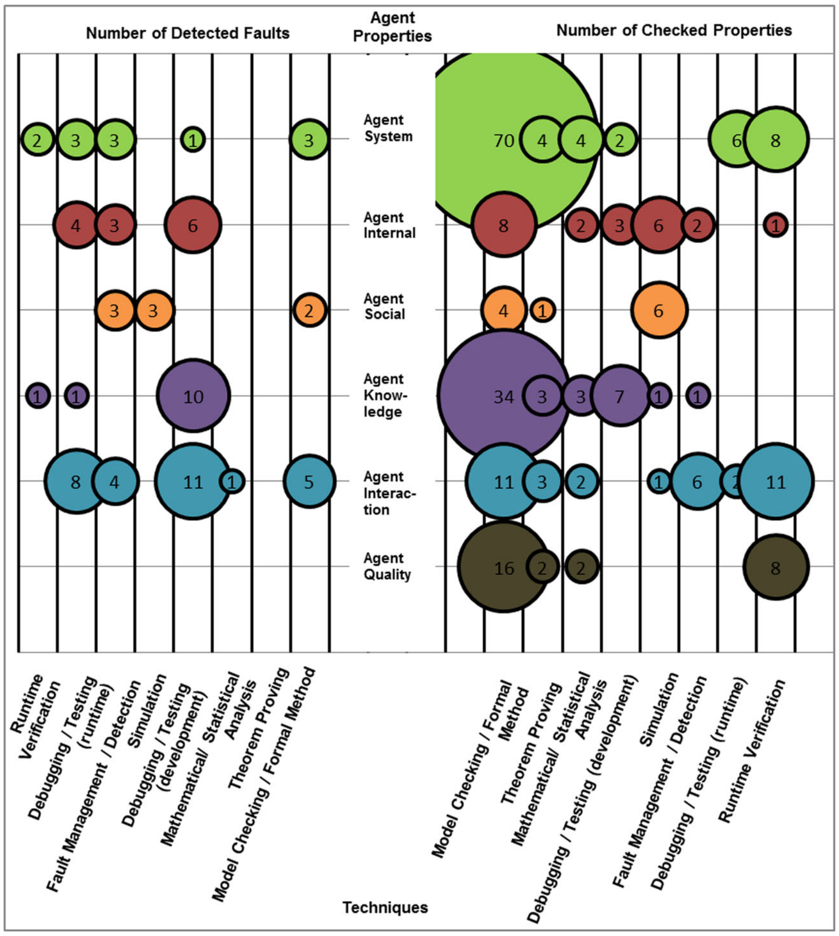

Fig. 12 The map of number of verification techniques with respect to the number of checked agent properties and detected faults

during development and 25% during runtime. The highest number of techniques used is the techniques are classified under formal method, model checking, and model-based verification techniques (44%) and followed by the testing and debugging during development (17%). The properties that are largely addressed by the selected studies are temporal properties (19%) and epistemic properties (9%). From this review, the findings are enlisted below:

(i) This study has shown that verification of agent and agent system properties have been significantly discussed in agent system and multi-agent system domain. However, most existing agent and agent system verification works have been focusing on checking properties using model checking and formal method techniques that were implemented during design level. Many verification techniques also have been discovered in this study including verification techniques performed during execution that are model-based fault detection, fault management, debugging and testing during runtime, and runtime verification. Some of these techniques have not been reviewed in the existing multi-agent systems and agent systems verification surveys. In this SLR, the works implementing those techniques are classified as the verification performed during runtime. The applicability of new techniques for verifying and validating agents and agent systems properties during execution especially in open environment where agents can freely join or leave the systems still needs to be further explored.

(ii) The importance of incorporating verification throughout the entire multi-agent systems or agent systems lifecycle is revealed. The classification of techniques shows that verification of agent and agent system properties have been performed not only during design to verify model and during development to test implementation, but also during runtime to verify agent and agent system execution properties. As compared to the design level, the number of agent and agent system verificationperformed during runtime is a lot lesser (refer to Figure 6). Thus, it might be the area that is worth to be explored by future researchers of agent systems verification.

(iii) By classifying the properties into six groups (system, internal, social, knowledge, interaction, and quality), it is easier for future researchers to situate their works into these groups. In order to perform and improve verification of certain agent characteristics in the future, the properties mentioned in this SLR should be considered by researchers. This classification also shows gaps in certain area of agent systems that need to be covered in the future. The coverage issues of agent and agent system verification have been previously discussed by [165] and [188]. Thus, by considering all properties, it will hopefully improve the coverage of the agent and agent system verification. The classification of agent properties based on agent requirement categories that are functional and non-functional properties indicates that non-functional properties are the areas that need further research.

(iv) The classification of the verification outcomes into checked properties satisfactions and detected properties violations (faults) also indicates that not many verification approaches focus to detect faults or properties violations (refer to Fig. 12). Bordini et al. [41] also stated that although agent applications has been verified to be correct, unpredictable emergent behavior can still occur due to agent reactiveness, proactiveness, flexibility, and social-ability characteristics. Thus, there is a need for future research to effectively identify agent properties violations such as faults and anomalies. From the mapping of the results, we identify that agent quality properties violations (faults) detection is yet to be explored. Some of the agent quality properties include safety, reliability, and trust.

(v) The relation between agent properties depends on the functional and non-functional requirements of the agent systems. At the end of this study, some of the properties are compatible. For example, temporal properties and epistemic properties are always verified as the combination of temporal and epistemic properties using temporal-epistemic logic (Belardinelli, 2015). Next, there is also a research that combines epistemic with strategy properties (Belardinelli, 2017) and deontic with trust properties (Osman, 2008). Detail review can be performed to study the relation between agent properties.

6 Conclusion

This paper aims to review the outcomes of the existing agent systems verification works, the techniques used to verify agents and agent system, and the addressed properties. First, the verification outcomes are grouped into the checked properties satisfaction and the detected properties violations. Second, the used techniques are the techniques for verification during design, during development, and during runtime. Finally, the addressed properties are classified into the properties of the agent system itself, the agent internal properties, the agent social properties, the agent knowledge properties, the agent interaction properties, and the agent quality properties. The main contributions of this work are the classification and mapping of the existing works based on these verification outcomes, the used techniques and the addressed properties in Fig. 12. The objectives of this work that are to find implementation gap and future research directions have been accomplished.

The limitation of this study is that the techniques and properties are classified into groups in order to accomplish the objectives of the study. In the future, further analysis needs to be performed in order to identify relationship between those techniques and the verified properties. By identifying the relationship between the techniques and properties, a hierarchical relationship map, or proper taxonomy between the techniques and properties can be produced.

Acknowledgements This study is supported by the Fundamental Research Grant Scheme (FRGS Vot: 4F904) that sponsored by Ministry of Higher Education (MOHE). Authors would like to thank Research Management Center (RMC) Universiti Teknologi Malaysia under Research University Funding Vot: 02G31, for the research activities in making this study a success.

Appendix A. Classification into verification outcomes

The literatures are classified into two verification outcomes i.e. properties satisfactions (correctness) checked and properties violations (faults) detected. Table 1 presents the works that address the properties of agent systems classified into approaches either for checking the properties satisfactions (correctness) or detecting the properties violations (faults). The table lists out the issues of the agent system itself that are commonly addressed by works in the area of agent and agent system verification. From the table, most works have been dedicated to perform agent systems temporal properties checking that is the main focus in agent systems verification research area and 4 works have detected faults related to temporal properties [116, 130, 206, 207]. Form all the works focusing on temporal properties, more than 30 works used model checking and other variations of model-based verification while the rest used theorem proving [79, 82], testing during development [68, 69] and runtime verification [66, 159]. Next, 11 works have been working on liveness properties of agent systems while 5 works managed to detect deadlock, livelock

Table 1 The works that check properties and detect faults of agent system properties

Agent System Properties	Checking Properties	Detecting Faults
Temporal properties	[11, 18, 21, 26–28, 36, 37, 39, 46, 66, 68, 69, 73, 78, 79, 82, 86, 88, 92–94, 102, 106, 109, 113, 115, 117, 123–126, 128, 142, 144, 147, 152, 159, 160, 169, 173, 194, 199, 200, 203, 211, 213, 221, 223]	[116, 130, 206, 207]
Event Pattern and Sequence	[44, 139, 157, 182]	[222, 229]
Probabilistic	[71, 108, 127]	
Concurrent Behavior	[213]	
Liveness	[5–7, 10, 20, 56, 66, 78, 80, 107, 126, 189]	[155, 156, 205, 230, 231]
Reachability	[14, 22, 78, 135, 167, 186, 211, 226]	
Fairness	[18, 78]	
Deontic	[86, 106, 136, 147, 184, 184]	

and bottleneck [154, 156, 205, 230, 231]. Other agent and agent system properties that are also important are events pattern and sequence, probabilistic, uncertainty and deterministic properties, system constraints, reachability, fairness, deontic, failure recovery, exceptional handling, trust and security properties as shown in Table 1.

The next properties group is the agent internal properties that include the properties of agent actions, characteristics and capabilities as individual (shown in Table 2). Three checked properties that are agent proactiveness and reactiveness [114], observability and controllability [85] and enforcement and regimentation [18] have been currently addressed. Three types of faults also have been detected that are delayed agent actions (6 works), agent plan-related faults (6 works) and malicious behavior (16 works).

When more than one agents work together to perform tasks, the agents share social properties that describe the behavior of the relation. Table 3 presents the third properties group that is the social properties of the agents addressed by existing verification of agent systems works. The checked social properties are cooperation, coordination and collaboration between agents [139, 175, 192, 201, 213], permission and commitments [78, 92–94, 98], self-organizing [25, 33, 84, 214] and scheduling [75]. The faults or violations of social properties that have been detected are commitments, deadline and conflict [101, 116].

Next, Table 4 presents the properties of agent knowledge addressed by the existing works. The most popular knowledge property is epistemic properties (23 works). This is followed by the BDI (belief, desire and intention) properties (10 works). Other properties in agent knowledge group are agent belief, agent goal, agent mental state and cognitive. Quite significant amount of works also have been dedicated to detect faults that mainly related to agent belief (8 works), goal (7 works) and role (1 work).

The following properties group is the agent interaction and communication properties. Agents interact to each other to perform tasks, to make decisions and to share resources by following certain language, protocol and ontology. The properties related to the interaction are shown in Table 5.

Finally, agent properties were classified into groups of agent quality properties. Agent quality includes agent non-functional properties that are safety, trust and security. Table 6 shows the agent quality properties.

Table 2 The works that check properties and detect faults of agent internal properties

Agent Internal Properties	Checking Properties	Detecting Faults
Proactiveness and Reactiveness	[114]	
Observability and Controllability	[85]	
Enforcement and Regimentation	[18]	
Delayed Actions		[64, 65, 161–164]
Plan-related Faults	[47]	[121, 165, 188, 198, 234, 235]
Malicious behavior		[17, 25, 34, 52, 53, 64, 99, 111, 134, 137, 170–172, 190, 224, 225]
Obligation	[216]	
Strategy	[29, 30, 55, 95]	
Role	[129]	[174]
Tasks Completion	[13]	

Table 3 The works that check properties and detect faults of agent social properties

Agent Social Properties	Checking Properties	Detecting Faults
Cooperation, Coordination and Collaboration	[139, 140, 175, 192, 213]	[120]
Adaptivity	[140]	
Satisfiability Relation	[67]	
Permission	[92–94]	
Commitments	[92–94]	[116]
Self-organizing	[25, 33, 84, 214]	
Conflict	[88]	[101, 141]
Scheduling	[129]	
Consensus or Agreement		[70]

Table 4 The works that check properties and detect faults of agent knowledge properties

Agent Knowledge Properties	Checking Properties	Detecting Faults
Epistemic Properties	[10, 21, 26–30, 46, 55, 72, 79, 98, 109, 113, 115, 124–126, 128, 130, 136, 143, 146–148, 152, 160, 200, 211, 221]	
BDI	[10, 17, 36, 37, 39, 68, 73, 127, 149, 223]	
Belief	[16, 39, 42, 43, 82, 127]	[64, 65, 150, 188, 234, 235]
Goal	[76, 107, 127, 136, 168]	[176–181, 183]
Mental State and Cognitive	[10, 50, 68, 136]	

Table 5 The works that check properties and detect faults of agent interaction properties

Agent Interaction Properties	Checking Properties	Detecting Faults
Interaction Pattern or Message Sequence	[139, 212, 220]	[3, 19, 64, 100, 118, 134, 166, 193, 218, 227–229]
Interaction Protocol or Rule	[8, 9, 15, 21, 48, 51, 90, 132, 133, 165, 184, 184]	[3, 63, 64, 83, 97, 187]
Message Content	[90, 206, 207]	[64]
Message Structure	[90]	
Message Exchange	[202]	[17, 81, 138, 196–198, 208, 234, 235]
Communication Language	[8, 153, 191]	
Interaction Ontology		[177, 178]
Interaction with Environment	[91]	
Resource Sharing	[107, 161, 162, 164]	[138]
Decision Making	[57–60, 74, 119, 209]	
Completeness of messages	[42, 43]	
Argumentation	[210]	
Information Spreading	[27]	

Table 6 The works that check properties and detect faults of agent quality properties

Agent Quality Properties	Checking Properties	Detecting Faults
Safety	[5–7, 10, 12, 20, 22, 54, 56, 66, 78, 80, 125, 126, 189]	
Trust	[24, 103, 184, 184]	
Security	[21, 124–126, 145, 204, 217]	

References

1. a Al N (2012) Conducting verification and validation of multi-agent systems. Int J Softw Eng Appl 3(5):115–124. ISSN 09762221
2. Bakar NA, Selamat A (2011) Towards implementing dynamic multi-agent V&V framework. In: The third software engineering postgraduates workshop (SEPoW 2011), JB, Malaysia
3. Abushark Y, Thangarajah J, Miller T, Harland J (2014) Checking consistency of agent designs against interaction protocols for early-phase defect location. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '14. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 933–940. ISBN 978-1-4503-2738-1
4. Philip A, Ali S, Roliana I, Mohd NM (2014) A systematic literature review of software requirements prioritization research. Inf Softw Technol 56(6):568–585. ISSN 09505849
5. Nadeem A (2015) Requirements, formal verification and model transformations of an agent-based system: a case study. arXiv:1501.05120, 5(3)
6. Nadeem A, Anique A (2014) Approach for the formal modeling of requirements, verification, and architecture of a multi-agent robotic system. IJCSI Int J Comput Sci Issue 11(2):237–246
7. Nadeem A, Aisha SG, Nadeem S (2014) Requirement analysis, Architectural design and Formal verification of a multi-agent based University Information Management System. International Journal of Computer Science & Information Technology (IJCSIT) 6(5):9
8. Alberti M, Gavanelli M, Lamma E, Chesani F, Mello P, Torroni P (2004) A logic based approach to interaction design in open multi-agent systems. In: Proceedings of the 13th IEEE international workshops on enabling technologies: Infrastructures for collaborative enterprises (WETICE-2004), 2nd international workshop "theory and practice of open computational systems (TAPOCS). IEEE Press, pp 387–392
9. Alberti M, Chesani F, Guerri A, Gavanelli M, Lamma E, Mello P, Milano M, Torroni P (2005) Expressing interaction in combinatorial auction through social integrity constraints. Proc of W(C)LP 2005-01:53–64
10. Natasha A, Mehdi D, Fahad K, Logan B, John J-JCh, Meyer JCh (2010) Using theorem proving to verify properties of agent programs. In: Dastani M, Hindriks KV, Meyer J-JC (eds) Specification and verification of multi-agent systems. Springer, US, pp 1–33. ISBN 9781441969835
11. Natasha A, Logan B, HoangNga NN, Abdur R (2010) Automated verification of resource requirements in multi-agent systems using abstraction. In: Ron van der M, Jan-Georg S (eds) International workshop on model checking and artificial intelligence, volume 6572 of lecture notes in computer science. Springer, Berlin, pp 69–84. ISBN 978-3-642-20673-3
12. Alotaibi H, Zedan H (2010) Runtime verification of safety properties in multi-agents systems. In: 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA), pp 356–362. ISBN 9781424481354
13. Aminof B, Murano A, Rubin S, Zuleger F (2016) Automatic verification of multi-agent systems in parameterised grid-environments. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems, pages 1190–1199. International Foundation for Autonomous Agents and Multiagent Systems
14. Ammar B, Abdallah K (2011) Towards the formal specification and verification of multi-agent based systems. International Journal of Computer Science Issues (IJCSI) 8(4):200–210
15. Ancona D, Briola D, El A, Seghrouchni F, Mascardi V, Taillibert P, Amal EFS, Viviana M, Davide A, Briola D, Taillibert P (2014) Efficient verification of MASs with projections. EMAS Pre-proceedings, pages 1–20
16. Anderson CA, Titler MG (2014) Development and verification of an agent-based model of opinion leadership. Implement Sci 9(1):136. ISSN 1748-5908
17. Vinay G, Ashok K (2012) EARCT-environment for automated rank based continuous agent testing. Int J Comput 11(3):180–190
18. Astefanoaei L, Dastani M, Meyer J-J, de Boer FS (2009) On the semantics and verification of normative multi-agent systems. J Universal Comput Sci 15(13):2629–2652. ISSN 09486968
19. Houhamdi Z, Athamena B, Athamena B, Houhamdi Z (2012) A petri net based multi-agent system behavioral testing. Mod Appl Sci 6(3):46–57. ISSN 19131844. https://doi.org/10.5539/mas.v6n3p46
20. Ayed LJB, Siala F (2009) Event-b based verification of interaction properties in multi-agent systems. JSW 4(4):357–364
21. Bagić M, Babac A, Kunštić M (2008) Verification of communication protocols in a multi-agent system. In: Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology, CSTST '08. ACM, New York, pp 286–291. ISBN 978-1-60558-046-3. https://doi.org/10.1145/1456223.1456283
22. Bakar NA, Selamat A (2011) Analyzing model checking approach for multi agent system verification. In: 2011 5th Malaysian conference in software engineering, MySEC 2011, pp 95–100. ISBN 9781457715310. https://doi.org/10.1109/MySEC.2011.6140650
23. Bakar NA, Selamat A (2012) Agent-based model checking verification framework. In: 2012 IEEE conference on open systems, ICOS 2012. ISBN 9781467310468
24. Bakar NA, Selamat A (2013) Runtime verification of multi-agent systems interaction quality. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 7802. LNAI, pp 435–444. ISBN 9783642365454
25. Barbosa J, Leitao P (2011) Simulation of multi-agent manufacturing systems using agent-based modelling platforms. In: 2011 9th IEEE International Conference on Industrial Informatics (INDIN), pp 477–482. https://doi.org/10.1109/INDIN.2011.6034926
26. Belardinelli F, Lomuscio A, Patrizi F (2014) Verification of agent-based artifact systems. J Artif Intell Res 51:333–376. ISSN 10769757. https://doi.org/10.1613/jair.4424
27. Belardinelli F, Grossi D, Lomuscio A (2015) Finite abstractions for the verification of epistemic properties in open multi-agent systems. In: IJCAI International Joint Conference on Artificial Intelligence, 2015-Janua (IJCAI), pp 854–860. ISSN 10450823
28. Belardinelli F, Kouvaros P, Lomuscio A (2017) Parameterised verification of data-aware multi-agent systems. In: Proceedings of the twenty-sixth International Joint Conference on Artificial Intelligence, {IJCAI-17}, pp 98–104. https://doi.org/10.24963/ijcai.2017/15
29. Belardinelli F, Lomuscio A, Murano A, Rubin S (2017) Verification of multi-agent systems with imperfect information and public actions. In: Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '17. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 1268–1276
30. Belardinelli F, Lomuscio A, Murano A, Rubin S (2017c) Verification of broadcasting multi-agent systems against an epistemic strategy logic. In: Proceedings of the twenty-sixth International Joint Conference on Artificial Intelligence, {IJCAI-17}, pp 91–97. https://doi.org/10.24963/ijcai.2017/14

31. Bellifemine F, Caire G, Greenwood D (2007) Developing Multi-Agent Systems with JADE. ISBN 9780470057476
32. Benerecetti M, Cimatti A (2003) Validation of multiagent systems by symbolic model checking. Springer, Berlin, pp 32–46. ISBN 978-3-540-36540-2
33. Bernon C, Gleizes M-P, Picard G (2007) Enhancing Self-organising emergent systems design with simulation. In: OâĂŽÄôHare GMP, Ricci A, OâĂŽÄôGrady M, Dikenelli O (eds) Engineering societies in the agents world vii, volume 4457 of lecture notes in computer science. Springer, Berlin, pp 284–299. ISBN 978-3-540-75522-7. https://doi.org/10.1007/978-3-540-75524-1_16
34. Berrada M, Bounabat B, Harti M (2007) Qualitative verification of multi-agents reactive decisional system using business process modeling notation. In: Proceedings - 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006 Main Conference Proceedings), IAT'06, pp 747–751. ISBN 9780769527482. https://doi.org/10.1109/IAT.2006.105
35. Bijani S, Robertson D (2012) A review of attacks and security approaches in open multi-agent systems. ISSN 02692821
36. Bordini RH, Fisher M, Wooldridge M, Visser W (2004) Model checking rational agents. IEEE Intell Syst 19(5):2004. ISSN 15411672. https://doi.org/10.1109/MIS.2004.47
37. Bordini RH, Fisher M, Visser W, Wooldridge M (2006) Verifying multi-agent programs by model checking, vol 12, p 2006. https://doi.org/10.1007/s10458-006-5955-7
38. Bordini RH, Hübner JF, Wooldridge M (2007) Programming Multi-Agent Systems in AgentSpeak using Jason. ISBN 9780470029008
39. Bordini RH, Dennis LA, Farwer B, Fisher M (2008) Automated verification of multi-agent programs. In: 23rd IEEE/ACM International Conference on Automated Software Engineering ASE 2008, pp 69–78. https://doi.org/10.1109/ASE.2008.17
40. Bordini RH, Dennis LA, Farwer B, Fisher M (2010) Directions for agent model checking*. In: Dastani M, Hindriks KV, Meyer J-JC (eds) Specification and verification of multi-agent systems. Springer, US, pp 103–123. ISBN 978-1-4419-6983-5
41. Bordini R, Dastani M, Winikoff M (2007) Current issues in multi-agent systems development
42. Bosse T, Jonker CM, van der Meij L, Sharpanskykh A, Treur J (2006) Specification and verification of dynamics in cognitive agent models. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2006. IAT '06, pp 247–254. https://doi.org/10.1109/IAT.2006.112
43. Bosse T, Lam DN, Barber KS (2008) Tools for analyzing intelligent agent systems. Web Intelli Agent Sys 6(4):355–371. ISSN 15701263. https://doi.org/10.3233/WIA-2008-0145
44. Bosse T, Jonker CM, van der Meij L, Sharpanskykh A, Treur J (2009) Specification and verification of dynamics in agent models. Int J Coop Inf Syst 18(01):167–193. ISSN 0218-8430. https://doi.org/10.1142/S0218843009001987
45. Botia JA, Gomez-Sanz JJ, Pavon J (2006) Intelligent data analysis for the verification of multi-agent systems interactions. In: Corchado E, Yin H, Botti V, Fyfe C (eds) International Conference on Intelligent Data Engineering and Automated Learning, volume 4224 of Lecture Notes in Computer Science. Springer, Berlin, pp 1207–1214. ISBN 978-3-540-45485-4
46. Bourahla M, Benmohamed M (2004) Formal specification and verification of multi-agent systems. Electronic notes in theoretical computer science. In: Proceedings of the 11th Workshop on Logic, Language, Information and Computation (WoLLIC 2004) Logic, Language, Information and Computation 2004, 123 (0): 5–17, 2005. ISSN 1571-0661. https://doi.org/10.1016/j.entcs.2004.04.042
47. Brahimi S, Maamri R, Sahnoun Z (2014) Validation and verification of agent and multi-agent plans in dynamic environment. CEUR Workshop Proc 1294:73–82. ISSN 16130073
48. Brazier FMT, Cornelissen F, Gustavsson R, Jonker CM, Lindeberg O, Polak B, Treur J (2004) Compositional verification of a multi-agent system for one-to-many negotiation. Appl Intell 20(2):95–117. ISSN 0924669X. https://doi.org/10.1023/B:APIN.0000013334.33853.0c
49. Bresciani P, Giorgini P, Mouratidis H, Manson G (2004) Multi-agent systems and security requirements analysis. pp. 1–13. https://doi.org/10.1007/b96018
50. Bulling N, Hindriks KV (2009) Towards a verification framework for communicating rational agents. Springer, Berlin, pp 177–182. ISBN 978-3-642-04143-3. https://doi.org/10.1007/978-3-642-04143-3_16
51. Caire G, Cossentino M, Negri A, Poggi A, Turci P (2004) Multi-agent systems implementation and testing. In: Fourth international symposium: from agent theory to agent implementation, pp 14–16
52. ÇakÄśrlar Ä, Gürcan Ö, Dikenelli O, Bora Å (2015) RatKit: Repeatable automated testing toolkit for agent-based modeling and simulation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9002, pp 17–27. ISBN 9783319146263. https://doi.org/10.1007/978-3-319-14627-0_2
53. Carrera A, Iglesias CA, Garijo M (2014) Beast methodology: an agile testing methodology for multi-agent systems based on behaviour driven development. Inf Syst Front 16(2):169–182. ISSN 1387-3326. https://doi.org/10.1007/s10796-013-9438-5
54. Casey W, Wright E, Morales JA, Appel M, Gennari J, Mishra B (2014) Agent-based trace learning in a recommendation-verification system for cybersecurity. In: Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference, pp 135–143. https://doi.org/10.1109/MALWARE.2014.6999404
55. Čermák P, Lomuscio A, Mogavero F, Murano A (2017) Practical Verification of Multi-Agent Systems against Slk Specifications. Information and Computation. ISSN 0890-5401
56. Chen J, Yang YL, Liu ZG (2013) Research on method of property verification of multi-agent system CGF architecture model. Appl Mech Mater 344:294–295. ISSN 16609336. https://doi.org/10.4028/www.scientific.net/AMM.344.294
57. Choi J, Tsourdos A (2011) Verification of decision making behaviour for heterogeneous multi-agent system. https://doi.org/10.2514/6.2011-6239
58. Choi J, Tsourdos A, White B (2010) Verification of multi-agent systems using formal approach. In: AIAA Guidance, Navigation, and Control Conference, pages —-. American Institute of Aeronautics and Astronautics. ISBN 978-1-60086-962-4. https://doi.org/10.2514/6.2010-8042
59. Choi J, Kim S, Tsourdos A (2013) Verification of heterogeneous multi-agent system using MCMAS. Int J Syst Sci 0(0):1–18. https://doi.org/10.1080/00207721.2013.793890
60. Choi J, Kim S, Tsourdos A (2015) Verification of heterogeneous multi-agent system using MCMAS. Int J Syst Sci 46(4):634–651. https://doi.org/10.1080/00207721.2013.793890
61. Cimatti A, Clarke E, Giunchiglia F, Roveri M (2000) NUSMV: a new symbolic model checker. Int J Softw Tools Technol Transfer 2(4):410–425. ISSN 14332779
62. Clarke EM, Grumberg O, Doron AP (1999) Model checking. The MIT Press, Cambridge
63. Coelho R, Kulesza U, von Staa A, Lucena C (2006) Unit testing in multi-agent systems using mock agents and aspects. In: Proceedings of the 2006 International Workshop on Software Engineering for Large-Scale Multi-Agent Systems - SELMAS

’06, SELMAS ’06. ACM, New York, pp 83–89. ISBN 1595933956. https://doi.org/10.1145/1138063.1138079
64. Coelho R, Cirilo E, Kulesza U, Von Staa A, Rashid A, Lucena C (2007) JAT: a test automation framework for multi-agent systems. In: IEEE International Conference on Software Maintenance, ICSM, pp 425–434. ISBN 1424412560. https://doi.org/10.1109/ICSM.2007.4362655
65. Collier R (2007) Debugging agents in agent factory. In: Bordini RH, Dastani M, Dix J, Seghrouchni A (eds) Programming multi-agent systems, volume 4411 of lecture notes in computer science. Springer, Berlin, pp 229–248. ISBN 978-3-540-71955-7. https://doi.org/10.1007/978-3-540-71956-4_14
66. Costantini S, De Gasperis G (2014) Runtime Self-Checking via Temporal (Meta-) Axioms for Assurance of Logical Agent Systems. pages 1–15
67. Da Silva P, De Melo ACV (2013) An approach for the verification of multi-agent systems by formally guided simulations. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol 2, pp 266–273. https://doi.org/10.1109/WI-IAT.2013.119
68. Dastani M, John J-JCh, Meyer JCh (2010) Correctness of multi-agent programs: a hybrid approach. In: Dastani M, Hindriks KV, Meyer J-JC (eds) Specification and verification of multi-agent systems. Springer, US, pp 161–194. ISBN 9781441969835
69. Dastani M, Brandsema J, Dubel A, John-Jules Ch, Meyer Ch (2009) Debugging bdi-based multi-agent programs. In: Braubach L, Briot J-P, Thangarajah J (eds) International workshop on programming multi-agent systems, volume 5919 of lecture notes in computer science. Springer, Berlin, pp 151–169. ISBN 978-3-642-14842-2. https://doi.org/10.1007/978-3-642-14843-9_10
70. Davoodi M, Meskin N, Khorasani K (2016) Simultaneous fault detection and consensus control design for a network of multi-agent systems. Automatica 66:185–194
71. Dekhtyar MI, Dikovsky AJ, Valiev M (2008) Temporal verification of probabilistic multi-agent systems. In: Avron A, Dershowitz N, Rabinovich A (eds) Pillars of computer science, volume 4800 of lecture notes in computer science. Springer, Berlin, pp 256–265. ISBN 978-3-540-78126-4. https://doi.org/10.1007/978-3-540-78127-1_14
72. Delgado C, Benevides M (2009) Verification of epistemic properties in probabilistic multi-agent systems. In: Braubach L, van der Hoek W, Petta P, Pokahr A (eds) Multiagent system technologies, volume 5774 of lecture notes in computer science. Springer, Berlin, pp 16–28. ISBN 978-3-642-04142-6. https://doi.org/10.1007/978-3-642-04143-3_3
73. Dennis LA, Farwer B, Bordini RHRH, Fisher M (2008) A flexible framework for verifying agent programs. In: Proceedings of the 7th International Conference on Autonomous Agent and Multiagent Systems, AAMAS ’08. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 1303–1306. ISBN 978-0-9817381-2-3
74. Dennis LA, Fisher M, Lincoln NK, Lisitsa A, Veres SM (2014) Practical verification of decision-making in agent-based autonomous systems. Autom Softw Eng 23(3):305–359. ISSN 1573-7535. https://doi.org/10.1007/s10515-014-0168-9
75. Earle CB, Fredlund LÅ, Moreno-Díaz R, Pichler F, Quesada-Arencibia A (2009) Debugging and verification of multi-agent systems. In: 12th International Conference Computer Aided Systems Theory - EUROCAST 2009, Las Palmas de Gran Canaria, Spain, February 15-20, 2009, Revised Selected Papers, volume 5717 of Lecture Notes in Computer Science, pp 263–270. ISBN 978-3-642-04771-8. https://doi.org/10.1007/978-3-642-04772-5
76. Ekinci EE, Tiryaki AM, Çetin Ö, Dikenelli O, âĹŽáetin â, Dikenelli O (2009) Goal-oriented agent testing revisited. In: Luck M, Gomez-Sanz J (eds) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 5386 of Lecture Notes in Computer Science. Springer, Berlin, pp 46–59. ISBN 978-3-642-01337-9. https://doi.org/10.1007/978-3-642-01338-6_13
77. El Fallah-Seghrouchni A, Gomez-Sanz JJ, Singh MP (2011) Formal methods in agent-oriented software engineering. Springer, Berlin, pp 213–228. ISBN 978-3-642-19208-1
78. El-Menshawy M, Bentahar J, Dssouli R (2010) Verifiable semantic model for agent interactions using social commitments. In: Dastani M, Segrouchni AEF, Leite J, Torroni P (eds) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 6039 LNAI of Lecture Notes in Computer Science. Springer, Berlin, pp 128–152. ISBN 978-3-642-13337-4. https://doi.org/10.1007/978-3-642-13338-1_8
79. Engelfriet J, Jonker CM, Treur J (1999) Compositional verification of multi-agent systems in temporal multi-epistemic logic. In: Intelligent Agents V: Agents Theories, Architectures, and Languages, vol 1555. Springer, pp 177–193. ISBN 978-3-540-65713-2
80. Esterline A, Gandluri B, Sundaresan M, Sankar J (2005) Verified models of multiagent systems for vehicle health management. In: Proceedings of SPIE, vol 5757, pp 602–613. https://doi.org/10.1117/12.600017
81. Fern L, Rodrigues O, De Carvalho GR, De Barros R, José C, Lucena P, Rodrigues L, De Carvalho GR, Paes R, Lucena C (2005) Towards an integration test architecture for open MAS. In: Proceedings of the 1st Workshop on Software Engineering for Agent-Oriented Systems/SBES
82. Fisher M, Wooldridge M (1997) On the formal specification and verification of multi-agent systems. Int J Coop Infor Syst 06(01):37–65. 10.1142/S0218843097000057
83. Flater D (2001) Debugging agent interactions: a case study. In: Proceedings of the 2001 ACM symposium on applied computing. ACM, ACM Press, pp 107–114
84. Fortino G, Garro A, Russo W, Caico R, Cossentino M, Termine F (2006) Simulation-driven development of multi-agent systems. In: Gentile A, Genco A, Sorce S (eds) Industrial simulation conference 2006 (ISC 2006), pages 17–24. EUROSIS (The European Simulation Society) & ETI (The European Technology Institute)
85. Franceschelli M, Martini S, Egerstedt M, Bicchi A, Giua A (2010) Observability and controllability verification in multi-agent systems through decentralized Laplacian spectrum estimation. In: 2010 49th IEEE Conference on Decision and Control (CDC), pp 5775–5780. ISBN 9781424477456. https://doi.org/10.1109/CDC.2010.5717400
86. Raimondi F, Lomuscio A (2004) Automatic verification of deontic properties of multi-agent systems. In: Proceedings of the 7th International Workshop on Deontic Logic in Computer Science (DEON 2004), Madeira, Portugal, May 26-28, 2004, number 3065/2004. Springer, pp 228–242. ISBN 3540221115 — 9783540221111
87. Gammie P, van der Meyden R (2004) MCK: model checking the logic of knowledge. Springer, Berlin, pp 479–483. ISBN 978-3-540-27813-9
88. Garanina NO, Bodin EV, Sidorova EA (2014) Using SPIN for verification of multi-agent data analysis. Model Anal Inform Sist 21(6):31–43. ISSN 1558-108X. https://doi.org/10.3103/S014641161507007X
89. Athanázio M, Gatti DC, Von Staa A (2006) Testing & Debugging Multi-Agent Systems: A State of the Art Report. p 28
90. German E, Sheremetov L (2008) An agent framework for processing fipa-acl messages based on interaction models. In: Luck M, Padgham L (eds) Agent-oriented software engineering

VIII, volume 4951 of Lecture Notes in Computer Science. Springer, Berlin, pp 88–102. ISBN 978-3-540-79487-5. https://doi.org/10.1007/978-3-540-79488-2_7
91. Giese H, Klein F (2007) Systematic verification of multi-agent systems based on rigorous executable specifications. Int J Agent-Oriented Softw Eng 1(1):28–62. ISSN 1746-1375. https://doi.org/10.1504/IJAOSE.2007.013264
92. Giordano L, Martelli A, Schwind C (2003) Specifying and verifying systems of communicating agents in a temporal action logic. In: Proceedings of the 8th conference of AI*IA, volume 2829 of LNAI. Springer, pp 262–274
93. Giordano L, Martelli A, Schwind Cx (2004) Verifying communicating agents by model checking in a temporal action logic. In: Proceedings Logics in artificial intelligence, 9th european conference, JELIA 2004. Springer, pp 57–69
94. Giordano L, Martelli A, Schwind C (2007) Specifying and verifying interaction protocols in a temporal action logic. J Appl Logic 5(2):214–234. ISSN 1570-8683. https://doi.org/10.1016/j.jal.2005.12.011
95. Perelli G (2015) Logics for multi-agent systems verification. PhD thesis, Universit‘ a degli studi di Napoli ”Federico II”
96. Góxmez-Sanz J, Gervais M-P, Weiss G (2004) A survey on agent-oriented oriented software engineering research. In: Methodologies and software engineering for agent systems, vol 11, pp 33–62. ISBN 978-1-4020-8058-6
97. Gomez-Sanz JJ, Botia J, Serrano E, Pavón J (2008) Testing and debugging of MAS interactions with INGENIAS. In: International Workshop on Agent-Oriented Software Engineering. Springer, pp 199–212
98. Guerin F, Pitt J (2002) Agent communication frameworks and verification. In: AAMAS 2002 Workshop on Agent Communication Languages, pp 200–202
99. Gurcan O, Dikenelli O, Bernon C (2011) Towards a generic testing framework for agent-based simulation models. In: 2011 Federated Conference on Computer Science and Information Systems (fedCSIS), pp 635–642
100. Gutiérrez C, García-Magariño I, Serrano E, Botía JA (2013) Robust design of multi-agent system interactions: a testing approach based on pattern matching. Eng Appl Artif Intell 26(9):2093–2104. ISSN 09521976. https://doi.org/10.1016/j.engappai.2013.06.006
101. Hamrouni N, Abderrahim Z (2012) Looking for verification and validation of a multi-agent system using new formalism: APN. In: 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp 30–35. https://doi.org/10.1109/SETIT.2012.6481885
102. Herd B, Miles S, McBurney P, Luck M (2014) Verification and validation of agent-based simulations using approximate model checking. Multi-Agent-Based Simul XIV(8235):53–70
103. Hnativ A, Ludwig SA (2009) Evaluation of trust in an ecommerce multi-agent system using fuzzy reasoning. In: IEEE International Conference on Fuzzy Systems, 2009. FUZZ-IEEE, pp 757–763. https://doi.org/10.1109/FUZZY.2009.5277344
104. Holzmann GJ (1997) The model checker SPIN. IEEE Trans Softw Eng 23(5):279–295. ISSN 00985589
105. Houhamdi Z (2011) Multi-agent system testing: a survey. Int J Advan Comput Sci Appl 2(6):135–141. ISSN 2158107X. https://doi.org/10.14569/IJACSA.2011.020620
106. Bing H, Ji G, Guo H (2009) Dynamic model of normative multi-agent system and its property verification mechanism. J Zhejiang University(engineering science) 43(6):1014–1019
107. Huang H-J, Wang X, Chen Q-C, Wang X-L (2005) Specification and verification of multi-agent systems with a property-preserving component-based methodology. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005, vol 1, pp 90–95. https://doi.org/10.1109/ICMLC.2005.1526925
108. Huget MP, Demazeau Y (2004) Evaluating multiagent systems: a record/replay approach. In: Proceedings - IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IAT 2004, pp 536–539. ISBN 0769521010. https://doi.org/10.1109/IAT.2004.1343013
109. Jamroga W, Penczek W (2012) Specification and verification of multi-agent systems. In: Bezhanishvili N, Goranko V (eds) Lectures on Logic and Computation, volume 7388 of Lecture Notes in Computer Science. Springer, Berlin, pp 210–263. ISBN 978-3-642-31484-1. https://doi.org/10.1007/978-3-642-31485-8_6
110. Janssen MA (2003) Complexity and ecosystem management: the theory and practice of multi-agent approaches. Complex Ecosyst Manag 8(2):63–74
111. Jiang Mx, Ding Z, Zhou M, Zhou Y (2015) Formal modeling and verification of secure mobile agent systems. In: 2015 IEEE international Conference on Automation Science and Engineering (CASE), pp 545–550
112. Johansson C, Bucanac C (1998) The v model, Crosstalk
113. Jones AV, Lomuscio A (2010) Distributed BDD-based BMC for the verification of multi-agent systems. In: van der Hoek W, Kaminka GA, Lespérance Y, Luck M, Sen S (eds) Transition, number Aamas. IFAAMAS, pp 675–682. ISBN 9781617387715
114. Jonker CM, Treur J (2002) Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. Int J Coop Inf Syst 11:51–91. ISBN 3540654933. https://doi.org/10.1142/S0218843002000480
115. Kacprzak M, Lomuscio A, Penczek W (2004) Verification of multiagent systems via unbounded model checking. In: Autonomous Agents and Multiagent Systems, pp 638–645. ISBN 1581138644. https://doi.org/10.1109/AAMAS.2004.296
116. Kafali Ö, Torroni P (2012) Exception diagnosis in multiagent contract executions. Ann Math Artif Intell 64(1):73–107. ISSN 10122443. https://doi.org/10.1007/s10472-012-9282-1
117. Kahloul L, Grira M (2014) Formal specification and verification of mobile agent systems. Int J Comput Commun 9(3):292–304. ISSN 1841-9836
118. Khamis MA, Nagi K (2013) Designing multi-agent unit tests using systematic test design patterns-(extended version). Eng Appl Artif Intell 26(9):2128–2142. ISSN 09521976. https://doi.org/10.1016/j.engappai.2013.04.009
119. Khattak AS, Sikander M, Khiyal H, Rizvi SS, Khiyal MSH, Rizvi SS, Sikander M, Khiyal H, Rizvi SS (2014) Verification & validation of a multi-agent meeting scheduling simulation model. Electron J Comput Sci Inf Technol 2(1):47–64
120. Khattak AS, Khiyal MSH, Rizvi SS (2015) Verification and validation of agent-based model using E-VOMAS approach. International Journal Of Computer Science and Network Security (IJCSNS) 15(3):29–35
121. Kissoum Y, Sahnoun Z (2007) A formal approach for functional and structural test case generation in multi-agent systems. In: 2007 IEEE/ACS International conference on computer systems and applications, pp 76–83. https://doi.org/10.1109/AICCSA.2007.370867
122. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Engineering 2:1051. ISSN 00010782
123. Klimek R, Faber Ł, Kisiel-Dorohinicki M (2014) Deduction-based modelling and verification of agent-based systems for data integration. Man-Machine Interact 3(242):361–368
124. Koleini M, Ryan M (2011) A knowledge-based verification method for dynamic access control policies. In: Qin S, Qiu Z (eds) Formal Methods and Software Engineering, volume 6991

of Lecture Notes in Computer Science. Springer, Berlin, pp 243–258. ISBN 978-3-642-24558-9. https://doi.org/10.1007/978-3-642-24559-6_18

125. Koleini M, Ritter E, Ryan M (2013) Model checking agent knowledge in dynamic access control policies. In: Piterman N, Smolka SA (eds) Tools and Algorithms for the Construction and Analysis of Systems, volume 7795 of Lecture Notes in Computer Science. Springer, Berlin, pp 448–462. ISBN 978-3-642-36741-0. https://doi.org/10.1007/978-3-642-36742-7_31
126. Koleini M, Ritter E, Ryan M (2014) Verification of agent knowledge in dynamic access control policies. CoRR, abs/1401.4: 448–462. https://doi.org/10.1007/978-3-642-36742-7_31
127. Konur S, Fisher M, Schewe S (2009) Verification of multi-agent systems via combined model checking
128. Kouvaros P (2013) Automatic verification of parameterised interleaved multi-agent systems. In: Proceedings of the 2013 international conference on autonomous agents and multi-agent systems, (Aamas), pp 861–868
129. Kouvaros P, Lomuscio A (2016) Parameterised verification for multi-agent systems. Artif Intell 234:152–189. ISSN 00043702. https://doi.org/10.1016/j.artint.2016.01.008
130. Kouvaros P, Lomuscio A (2017) Parameterised verification of infinite state multi-agent systems via predicate abstraction. Proceedings of the 31th Conference on Artificial Intelligence (AAAI) 2017:3013–3020
131. Kwiatkowska M, Norman G, Parker D (2011) PRISM 4.0: Verification of probabilistic real-time systems. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 6806 LNCS, pp 585–591. ISBN 9783642221095
132. Lacey T, Deloach SA (2000) Automatic verification of multiagent conversations. In: Proceedings of the Eleventh Annual Midwest Artificial Intelligence and Cognitive Science Conference. AAAI Press, pp 93–100
133. Lacey TH, Deloach SA (2000) Verification of agent behavioral models. In: Proceedings of the international conference on artificial intelligence (IC-AI'2000). CSREA Press, pp 557–564
134. Lam DN, Barber KS (2005) Debugging agent behavior in an implemented agent system. In: Bordini RH, Dastani M, Dix J, Seghrouchni AEF (eds) Programming multi-agent systems, volume 3346 of lecture notes in computer science. Springer, Berlin, pp 104–125. ISBN 978-3-540-24559-9. https://doi.org/10.1007/978-3-540-32260-3_6
135. Latif N, Hassan M, Hasan M (2011) Formal verification for interaction protocol in agent-based e-learning system using model checking toolkit - MCMAS. In: Zain JM, Wan Mohd WM, El-Qawasmeh E (eds) Software engineering and computer systems, volume 180 of communications in computer and information science. Springer, Berlin, pp 412–426. ISBN 978-3-642-22190-3. https://doi.org/10.1007/978-3-642-22191-0_36
136. Laureano-Cruces AL, Barceló-Aspeitia AA (2003) Formal verification of multi-agent systems behaviour emerging from cognitive task analysis. J Exp Theor Artif Intell 15(4):407–431. ISSN 0952-813X. https://doi.org/10.1080/0952813031000119719
137. Lee P, Saleh O, Alomair B, Bushnell L, Poovendran R (2014) Graph-based verification and misbehavior detection in multi-agent networks. In: Proceedings of the 3rd International Conference on High Confidence Networked Systems, HiCoNS'14. ACM, New York, pp 77–84. ISBN 978-1-4503-2652-0. https://doi.org/10.1145/2566468.2566477
138. Li Y, Fang H, Chen J, Shang C (2016) Distributed fault detection and isolation for multi-agent systems using relative information. In: American control conference (ACC), 2016. American Automatic Control Council (AACC), pp 5939–5944
139. Van Liedekerke MH, Avouris NM (1995) Debugging multi-agent systems. Inf Softw Technol 37:102–112
140. Lim YJ, Hong G, Shin D, Jee E, Bae D-H (2016) A runtime verification framework for dynamically adaptive multi-agent systems. In: 2016 International conference on big data and smart computing (bigcomp). IEEE, pp 509–512
141. Liu X, Gao X, Han J (2015) Fault detection for high-order multi-agent systems with disturbances. In: The 27th Chinese Control and Decision Conference (2015 CCDC), pp 3814–3819. https://doi.org/10.1109/CCDC.2015.7162590
142. Lomuscio A, Michaliszyn J (2014) An abstraction technique for the verification of multi-agent systems against ATL specifications. In: Association for the advancement of artificial intelligence. ISBN 9781577356578
143. Lomuscio A, Michliszyn J (2016) Verification of multi-agent systems via predicate abstraction against ATLK specifications. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. International Foundation for Autonomous Agents and Multiagent Systemsx, pp 662–670
144. Lomuscio A, Paquet H (2015) Verification of multi-agent systems via sdd-based model checking. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, pages 1713–1714, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6
145. Lomuscio A, Penczek W (2007) Model checking security protocols: a multiagent systems approach. In: Proceedings of the 10th International Workshop on con-currency, specification and programming. Warsaw University Press, Poland, pp 400–412
146. Lomuscio A, Solanki M (2009) Mapping OWL-S processes to multi agent systems: a verification oriented approach. In: International Conference on Advanced Information Networking and Applications Workshops, 2009. WAINA '09, pp 488–493. ISBN 9780769536392. https://doi.org/10.1109/WAINA.2009.52
147. Lomuscio A, Qu H, Raimondi F (2009) MCMAS: a model checker for the verification of multi-agent systems. In: Bouajjani A, Maler O (eds) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 5643 of Lecture Notes in Computer Science. Springer, Berlin, pp 682–688. ISBN 978-3-642-02657-7
148. Lomuscio A, Hongyang Q, Raimondi F (2015) MCMAS: an open-source model checker for the verification of multi-agent systems. Int J Softw Tools Technol Transfer:1–22
149. Low CKK, Chen TY, RâĹŽâL'ěnnquist R, Rónnquist R (1999) Automated test case generation for BDI agents. Auton Agent Multi-Agent Syst 2(4):311–332. ISSN 13872532. https://doi.org/10.1023/A:1010011219782
150. Passos LÍS, Rossetti RJF, Gabriel J (2013) Diagnosis of unwanted behaviours in multi-agent systems. In: 11Th european workshop on multi-agent systems (EUMAS 2013). France, Toulouse
151. Luck M, McBurney P, Preist C (2003) Agent technology: enabling next generation computing (A Roadmap for Agent Based Computing)
152. Luo X, Su K, Sattar A, Reynolds M (2006) Verification of multi-agent systems via bounded model checking. In: Sattar A, Kang B-h (eds) AI 2006: advances in artificial intelligence, volume 4304 of lecture notes in computer science. Springer, Berlin, pp 69–78. ISBN 978-3-540-49787-5. https://doi.org/10.1007/11941439_11
153. Luo Xiangyu, Zou Mengmeng, Luo Lingjie (2012) A modeling and verification method to multi-agent systems based on KQML Electrical electronics engineering (EEESYM), 2012 IEEE symposium on, pages 690–693, jun, https://doi.org/10.1109/EEESym.2012.6258753

154. Mani N (2008) Testing and monitoring multi-agent systems for deadlock detection. PhD thesis. http://www.softqual.ucalgary.ca/pubs/theses/2009_Nariman_Mani_Thesis.pdf
155. Mani N, Garousi V, Far BH (2008) Testing multi-agent systems for deadlock detection based on UML models. In: Canadian Conference on Electrical and Computer Engineering, Boston, USA, pp 77–84. ISBN 9781424416431. https://doi.org/10.1109/CCECE.2008.4564814
156. Mani N, Garousi V, Far BH (2010) Search-based testing of multi-agent manufacturing systems for deadlocks based on models. Int J Artif Intell Tools 19(04):417–437. https://doi.org/10.1142/S0218213010000261
157. Marzougui B, Hassine K, Barkaoui K (2011) Method for verification of a multi agents system. In: 2011 Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pp 62–65. https://doi.org/10.1109/ISMS.2011.21
158. Mehmood A, Jawawi DNA (2013) Aspect-oriented model-driven code generation: a systematic mapping study. In: Information and software technology, vol 55, pp 395–411. ISBN 0950-5849
159. Meron D, Mermet B (2007) A tool architecture to verify properties of multiagent system at runtime. In: Bordini RH, Dastani M, Dix J, Seghrouchni AEF (eds) Programming multi-agent systems, volume 4411 of lecture notes in computer science. Springer, Berlin, pp 201–216. ISBN 978-3-540-71955-7. https://doi.org/10.1007/978-3-540-71956-4_12
160. Męski A, Penczek W, Szreter M, Woźna-Szcześniak B, Zbrzezny A (2014) BDD-versus SAT-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance. Auton Agent Multi-Agent Syst 28(4):558–604. ISSN 13872532. https://doi.org/10.1007/s10458-013-9232-2
161. Micalizio R, Torasso P, Torta G (2003) The architecture of the diagnostic agent in the ROBOCARE project: the long report. Technical report Technical Report RC-TR-1203
162. Micalizio R, Torasso P, Torta G (2004) On-line monitoring and diagnosis of multi-agent systems: a model based approach. In: ECAI, pp 848–852
163. Micalizio R, Torasso P, Torta G (2006) On-line monitoring and diagnosis of a team of service robots: a model-based approach. AI Commun 19(4):313–340. ISSN 0921-7126
164. Micalizio R, Torasso P, Torta G (2006) Synthesizing diagnostic explanations from monitoring data in multi-robot systems. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, AIA 2006, AIA'06. ACTA Press, Anaheim, pp 279–286. ISBN 0-88986-556-6
165. Miller T, Padgham L, Thangarajah J (2011) Test coverage criteria for agent interaction testing. In: Weyns D, Gleizes M-P (eds) Agent-oriented software engineering XI, volume 6788 of lecture notes in computer science. Springer, Berlin, pp 91–105. ISBN 978-3-642-22635-9
166. Mireslami S, Far BH (2013) Automated verification of AUML based multi-agent system design. In: Canadian Conference on Electrical and Computer Engineering, pp 1–4. ISBN 9781479900329. https://doi.org/10.1109/CCECE.2013.6567818
167. Mohammed A, Furbach U (2009) Multi-agent systems: modeling and verification using hybrid automata. In: Braubach L, Briot J-P, Thangarajah J (eds) International workshop on programming multi-agent systems, volume 5919 of lecture notes in computer science. Springer, Berlin, pp 49–66. ISBN 978-3-642-14842-2. https://doi.org/10.1007/978-3-642-14843-9_4
168. Moreno M, Pavón J, Rosete A (2009) Testing in agent oriented methodologies. Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5518 LNCS (PART 2):138–145. ISSN 03029743
169. Moscato F, Venticinque S, Aversa R, Di Martino B (2008) Formal modeling and verification of real-time multi-agent systems: the REMM framework. In: Badica C, Mangioni G, Carchiolo V, Burdescu D (eds) Intelligent distributed computing, systems and applications, volume 162 of studies in computational intelligence. Springer, Berlin, pp 187–196. ISBN 978-3-540-85256-8. https://doi.org/10.1007/978-3-540-85257-5_19
170. Moshirpour M, Mousavi A, Far BH (2010) Model based detection of implied scenarios in multi agent systems. In: 2010 IEEE International Conference on Information Reuse and Integration (IRI), pp 63–68. https://doi.org/10.1109/IRI.2010.5558962
171. Moshirpour M, Mousavi A, Far BH (2012) Detecting emergent behavior in distributed systems using scenario-based specifications. Int J Softw Eng Knowl Eng 22(6):729–746. https://doi.org/10.1142/S0218194012400104
172. Moshirpour M, Mani N, Eberlein A, Far BH (2013) Model based approach to detect emergent behavior in multi-agent systems. In: 12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013, volume 2 of AAMAS '13. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 1285–1286. ISBN 978-1-4503-1993-5
173. Murano A, Perelli G, Federico N (2015) Pushdown multi-agent system verification. Number Ijcai, 1090–1096
174. Vivekanandan K, Sivakumar N (2012) Agent oriented software testing – role oriented approach. Int J Adv Comput Sci Appl (IJACSA) 3(12):156–163
175. Ndumu DT, Nwana HS, Lee LC, Collis JC (1999) Visualising and debugging distributed multi-agent systems. In: Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS '99. ACM, New York, pp 326–333. ISBN 1-58113-066-X. https://doi.org/10.1145/301136.301220
176. Cu D, Perini A, Tonella P, Kessler FB (2007) Nguyen automated continuous testing of multi-agent systems. In: Fifth European Workshop on Multi-Agent Systems (EUMAS)
177. Nguyen DC, Perini A, Tonella P (2008) A goal-oriented software testing methodology. In: Luck M, Padgham L (eds) Agent-oriented software engineering VIII, volume 4951 of lecture notes in computer science. Springer, Berlin, pp 58–72. ISBN 978-3-540-79487-5. https://doi.org/10.1007/978-3-540-79488-2_5
178. Nguyen CDD, Perini A, Tonella P (2009) Experimental evaluation of ontology-based test generation for multi-agent systems. In: Luck M, Gomez-Sanz JJ (eds) Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 5386 of lecture notes in computer science. Springer, Berlin, pp 187–198. ISBN 9783642013379. https://doi.org/10.1007/978-3-642-01338-6_14
179. Nguyen CD, Perini A, Tonella P (2010) Goal-oriented testing for MASs. Int J Agent-Oriented Softw Eng 4(1):79. ISSN 1746-1375. https://doi.org/10.1504/IJAOSE.2010.029810
180. Nguyen CD, Perini A, Tonella P (2010) Goal-oriented Testing for MASs. Int J Agent-Oriented Softw Eng 4(1):79–109. ISSN 1746-1375. https://doi.org/10.1504/IJAOSE.2010.029810
181. Nguyen CD, Perini A, Bernon C, Pavón J, Thangarajah J (2011) Testing in multi-agent systems. In: Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 6038 LNCS, pp 180–190. ISBN 9783642192074
182. Nugraheni CE (2011) Formal verification of parameterized multi-agent systems using predicate diagrams*. In: Computation tools 2011: the second international conference on computational logics, algebras, programming, tools, and benchmarking, Rome

183. Núñez M, Rodríguez I, Rubio F (2005) Specification and testing of autonomous agents in e-commerce systems. Softw Test Verification Reliab 15(4):211–233. ISSN 1099-1689. https://doi.org/10.1002/stvr.323
184. Osman N, Robertson D, Walton C (2006) Run-time model checking of interaction and deontic models for multi-agent systems. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '06. ACM, New York, pp 238–240. ISBN 1-59593-303-4. https://doi.org/10.1145/1160633.1160674
185. Osman NZ (2008) Runtime verification of deontic and trust models in multiagent interactions. PhD thesis, University of Edinburgh
186. Ou-Yang C, Juan Y-C (2010) Applying process mining approach to support the verification of a multi-agent system. J Syst Sci Syst Eng 19(2):131–149. ISSN 1004-3756. https://doi.org/10.1007/s11518-010-5132-z
187. Padgham L, Winikoff M, Poutakidis D (2005) Adding debugging support to the Prometheus methodology. Eng Appl Artif Intell 18(2):173–190. ISSN 09521976. https://doi.org/10.1016/j.engappai.2004.11.018
188. Padgham L, Zhang Z, Thangarajah J, Miller T (2013) Model-based test oracle generation for automated unit testing of agent systems. IEEE Trans Softw Eng 39(9):1230–1244. ISSN 00985589
189. Pallottino L, Scordio VG, Frazzoli E, Bicchi A (2006) Probabilistic verification of a decentralized policy for conflict resolution in multi-agent systems. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA, pp 2448–2453. ISBN 0780395069. https://doi.org/10.1109/ROBOT.2006.1642069
190. Passos LS, Abreu R, Rossetti RJF (2015) Spectrum-based fault localisation for multi-agent systems. IJCAI International Joint Conference on Artificial Intelligence, 2015-Janua (Ijcai): 1134–1140. ISSN 10450823
191. Paurobally S, Cunningham J, Jennings NR (2004) Verifying the contract net protocol: a case study in interaction protocol and agent communication semantics. In: 2Nd International Workshop on Logic and Communication in Multi-Agent Systems, pp 98–117
192. Penczek W, Lomuscio A (2003) Verifying epistemic properties of multi-agent systems via bounded model checking. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS '03, p 209. ISSN 01692968. https://doi.org/10.1145/860575.860609
193. Peng W, Krueger W, Grushin A, Carlos P, Manikonda V, Santos M (2009) Graph-based methods for the analysis of large-scale multiagent systems. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '09. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 545–552. ISBN 978-0-9817381-6-1
194. Pilecki J, Bednarczyk MA, Jamroga W (2014) Synthesis and verification of uniform strategies for multi-agent systems. Comput Logic Multi-Agent Syst 8624:166–182
195. Potiron K, Taillibert P, Seghrouchni AEF (2008) A step towards fault tolerance for multi-agent systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5118 LNAI: 156–172. ISSN 03029743
196. Poutakidis D, Padgham L, Winikoff M (2002) Debugging multi-agent systems using design artifacts: the case of interaction protocols. In: Proceedings of the first international joint conference on autonomous agents and multi-agent systems, pp 960–967
197. Poutakidis D, Padgham L, Winikoff M (2003) An exploration of bugs and debugging in multi-agent systems. In: International symposium on methodologies for intelligent systems. Springer, pp 628–632
198. Poutakidis D, Winikoff M, Padgham L, Zhang Z (2009) Debugging and testing of multi-agent systems using design artefacts. Springer, US. ISBN 9780387892986
199. Qasim A, Asad S, Kazmi R, Fakhir I (2015) Formal specification and verification of real-time multi-agent systems using timed-arc petri nets. Advan Electr Comput Eng 15(3):73–78
200. Raimondi F, Pecheur C, Lomuscio A (2005) Applications of model checking for multi-agent systems: verification of diagnosability and recoverability. In: Proceedings of concurrency, specification & programming (CS&p), Warsaw University, pp 433–444
201. Röhl M, Uhrmacher A (2004) Controlled experimentation with agents - models and implementations. In: Gleizes MP, Omicini A, Zambonelli F (eds) ESAW, volume 3451 of Lecture Notes in Computer Science. Springer, pp 292–304. ISBN 3-540-27330-1
202. Rouff C (2002) A test agent for testing agents and their communities. In: IEEE Aerospace Conference Proceedings, vol 5, pp 2633–2638. ISBN 078037231X. https://doi.org/10.1109/AERO.2002.1035446
203. Roungroongsom C, Pradubsuwun D (2015) Formal verification of multi-agent system based on JADE A semi-runtime approach. Recent Advan Inf Commun Technol 361:297–306
204. Sabri KE, Khedri R, Jaskolka J (2009) Verification of information flow in agent-based systems. In: Babin G, Kropf P, Weiss M (eds) Lecture notes in business information processing, volume 26 of lecture notes in business information processing. Springer, Berlin, pp 252–266. ISBN 9783642011863. https://doi.org/10.1007/978-3-642-01187-0_22
205. Salamon T (2009) A three-layer approach to testing of multi-agent systems. In: Papadopoulos GA, Wojtkowski W, Wojtkowski G, Wrycza S, Zupancic J (eds) Information systems development. Springer, pp 393–401. ISBN 978-0-387-84809-9. https://doi.org/10.1007/b137171_41
206. Selamat A, Dianah S, Bujang A (2008) The design of model checking agent for SMS management system. In: Nguyen NT, Jo GS, Howlett RJ, Jain LC (eds) Agent and multi-agent systems: technologies and applications, volume 4953 of lecture notes in computer science. Springer, Berlin, pp 813–821. ISBN 978-3-540-78581-1
207. Selamat A, Lockman MTT (2009) Multi-agent verification of RFID system. In: Nguyen NT, Katarzyniak RP, Janiak A (eds) New challenges in computational collective intelligence, volume 244 of studies in computational intelligence. Springer, Berlin, pp 255–268. ISBN 9783642039577
208. Serrano E, Muñoz A, Botia J (2012) An approach to debug interactions in multi-agent system software tests. Inf Sci 205:38–57. ISSN 0020-0255. https://doi.org/10.1016/j.ins.2012.04.001
209. Serrano E, Rovatsos M, Botía JA, Botía JA (2013) Data mining agent conversations: a qualitative approach to multiagent systems analysis. Inf Sci 230(0):132–146. ISSN 0020-0255. https://doi.org/https://doi.org/10.1016/j.ins.2012.12.019
210. Shetty S, Kiran HSS, Namala MB, Singh S (2011) Logical modeling and verification of a strength based multi-agent argumentation scheme using NuSMV. In: Meghanathan N, Kaushik BK, Nagamalai D (eds) Advances in computer science and information technology, volume 131 of communications in computer and information science. Springer, Berlin, pp 378–387. ISBN 978-3-642-17856-6. https://doi.org/10.1007/978-3-642-17857-3_38
211. Song S, Hao J, Liu Y, Sun J, Leung HF, Dong JS (2012) Analyzing multi-agent systems with probabilistic model checking

approach. In: 2012 34th International Conference on Software Engineering (ICSE), pp 1337–1340. ISBN 9781467310673. https://doi.org/10.1109/ICSE.2012.6227085

212. Srivastava PR, Karthik Anand V, Subba Reddy V, Raghurama G (2008) Regression testing techniques for agent oriented software. In: International Conference on Information Technology, 2008. ICIT '08, pp 221–225. ISBN 978-0-7695-3513-5. https://doi.org/10.1109/ICIT.2008.30
213. Stolzenburg F, Arai T (2003) From the specification of multiagent systems by statecharts to their formal analysis by model checking: towards safety-critical applications. In: Schillo M, Klusch M, Muller J, Tianfield H (eds) Multiagent System Technologies, volume 2831 of Lecture Notes in Computer Science. Springer, Berlin, pp 131–143. ISBN 978-3-540-20124-3
214. Sudeikat J, Renz W (2009) A systemic approach to the validation of SelfâĂŽÀìOrganizing Dynamics within MAS. In: Luck M, Gomez-Sanz JJ (eds) Agent-oriented software engineering IX, volume 5386 of lecture notes in computer science. Springer, Berlin, pp 31–45. ISBN 978-3-642-01337-9. https://doi.org/10.1007/978-3-642-01338-6_3
215. Sycara KP (1998) Multiagent systems. AI Mag 19(2):79. ISSN 0738-4602
216. Tao Z, Hong X, Shao-Bin H (2015) A formal verification method of obligation policy in multi-agent system international journal of u- and e- service. Sci Technol 8(11):113–124
217. Tekbacak F, Tuglular T, Dikenelli O (2009) An architecture for verification of access control policies with multi agent system ontologies. In: 33rd Annual IEEE International of Computer Software and Applications Conference, 2009. COMPSAC '09, vol 2, pp 52–55. https://doi.org/10.1109/COMPSAC.2009.114
218. Tiryaki AM, âĹŽñztuna S, Dikenelli O, Erdur RC, Öztuna S (2007) SUNIT: a unit testing framework for test driven development of multi-agent systems. In: Padgham L, Zambonelli F (eds) Agent-oriented software engineering VII, volume 4405 of lecture notes in computer science. Springer, Berlin, pp 156–173. ISBN 978-3-540-70944-2. https://doi.org/10.1007/978-3-540-70945-9_10
219. UPPAAL Team (2017) UPPAAL. http://www.uppaal.org/
220. Van Eijk RM, de Boer FS, van der Hoek W, Meyer JJ (2003) A verification framework for agent communication. Auton Agent Multi-Agent Syst V6(2):185–219. https://doi.org/10.1023/A:1021836202093
221. Wan W, Bentahar J, Hamza AB (2011) Model checking epistemic and probabilistic properties of multi-agent systems. In: Mehrotra KG, Mohan CK, Oh JC, Varshney PK, Ali M (eds) Modern approaches in applied intelligence, volume 6704 of lecture notes in computer science. Springer, Berlin, pp 68–78. ISBN 978-3-642-21826-2. https://doi.org/10.1007/978-3-642-21827-9_8
222. Wang S, Zhu H (2012) CATest: a test automation framework for multi-agent systems. In: 2012 IEEE 36th Annual of Computer Software and Applications Conference (COMPSAC), pp 148–157. ISBN 9780769547367. https://doi.org/10.1109/COMPSAC.2012.24
223. Wooldridge M, Fisher M, Huget M-P, Parsons S (2002) Model checking multi-agent systems with MABLE. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02. ACM, New York, p 952. ISBN 1581134800
224. Wright CJ, Mcminn P, Gallardo J (2012) Testing multi-agent based simulations using MASTER. In: Multi-agent based simulation 2012
225. Wright CJ, McMinn P, Gallardo J (2013) Towards the auto matic identification of faulty multi-agent based simulation runs using MASTER. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 7838 LNAI, pp 143–156. ISBN 9783642388583. https://doi.org/10.1007/978-3-642-38859-0_11
226. Dianxiang X, Volz RA, Ioerger TR, Yen J (2002) Modeling and verifying multi-agent behaviors using predicate transition nets. In: Proceedings of the 14th international conference on software engineering and knowledge engineering. ACM Press, pp 193–200
227. Xu P, Deters R (2004) Using event-streams for fault-management in MAS. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004. (IAT 2004). Proceedings, pp 433–436. ISBN 0769521010. https://doi.org/10.1109/IAT.2004.1342989
228. Xu P, Deters R (2004) MAS and fault-management. In: Proceedings of the 2004 International Symposium on Applications and the Internet. IEEE, pp 283–286. ISBN 0-7695-2068-5. https://doi.org/10.1109/SAINT.2004.1266129
229. Peng X, Deters R (2005) Fault-management for multi-agent systems. In: Proceedings of the 2005 Symposium on Applications and the internet, pp 287–293. https://doi.org/10.1109/SAINT.2005.29
230. Yeung WL (2011) Behavioral modeling and verification of multi-agent systems for manufacturing control. Expert Syst Appl 38(11):13555–13562. ISSN 09574174. https://doi.org/10.1016/j.eswa.2011.04.067
231. Yeung WL (2011) Formal verification of negotiation protocols for multi-agent manufacturing systems. Int J Prod Res 49(12):3669–3690. ISSN 00207543. https://doi.org/10.1080/00207543.2010.492407
232. Yu E, Cysneiros LM (2003) Designing for privacy in a multi-agent world. Lect Note Artif Intell (Subseries of Lecture Notes in Computer Science) 2631:209–223. ISSN 03029743
233. Zamani A (2011) Evaluation of the testing methods in agent-oriented software engineering. http://worldcomp-proceedings.com/proc/p2011/SER3075.pdf
234. Zhang Z, Thangarajah J, Padgham L (2007) Automated unit testing for agent systems. In: Proceedings of the 2nd international working conference on evaluation of novel approaches to software engineering (ENASE), Spain, pp 10–18
235. Zhang Z, Thangarajah J, Padgham L (2008) Automated unit testing intelligent agents in PDT. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and multi-agent systems: Demo Papers, AAMAS '08. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 1673–1674

Najwa Abu Bakar is currently completing her PhD studies at Universiti Teknologi Malaysia. Previously she had completed her Master at Universiti Sains Malaysia and BSc Degree in Computer Science, Minor in Mathematics from University of Idaho, USA. Her research interests include multi-agent verifications, intrusion detections, computer security.

Ali Selamat is a professor at Faculty of Computing, Universiti Teknologi Malaysia, Malaysia. He is currently a Chief Information Officer (CIO) and Director of Center for Information and Communication Technology, Universiti Teknologi Malaysia (UTM). He was the Dean of Research, Knowledge Economy Research Alliance, UTM. He is currently a Chair of IEEE Computer Society Malaysia Chapter. He is also the Editorial Boards of Knowledge Based Systems Elsevier, International Journal of Information and Database Systems (IJJIDS), Inderscience Publications, Vietnam Journal of Computer Science, Springer Publications. He was the Program Chair of International Conference on Software Engineering Tools, Methodologies (SOMET), 2017, Kanazawa, Japan. His research interests include cloud based software engineering, software agents, information retrievals, pattern recognitions, genetic algorithms, neural networks and soft-computing, knowledge management, key performance indicators.