A Survey of Self-organisation Mechanisms in Multi-Agent Systems

Dayong Ye, Minjie Zhang and Athanasios V. Vasilakos

***Abstract*—This paper surveys the literature over a period of the last decades in the field of self-organising multi-agent systems. Self-organisation has been extensively studied and applied in multi-agent systems and other fields, e.g., sensor networks and grid systems. Self-organisation mechanisms in other fields have been thoroughly surveyed. However, there has not been a survey of self-organisation mechanisms developed for use in multi-agent systems. The aim of this paper is to provide a survey of existing literature on self-organisation mechanisms in multi-agent systems. This paper also highlights the future work on the key research issues in multi-agent systems. This paper serves as a guide and a starting point for anyone who will conduct research on self-organisation in multi-agent systems. Also, this paper complements existing survey studies of self-organisation in multi-agent systems.**

Keywords - Self-organisation, Multi-Agent Systems, Distributed Artificial Intelligence

I. Introduction

A. Multi-agent systems

Most research in artificial intelligence to date has dealt with developing theories, techniques and systems to study and understand the behaviour and reasoning properties of a single cognitive entity, i.e., an agent [1]. Agent-based system technology has generated much excitement in recent years because of its promise as a new paradigm for conceptualising, designing and implementing software systems. The capacity of a single agent is limited by its knowledge, its computing resources and its perspectives. This bounded rationality [2] is one of the underlying reasons for creating problem-solving organisations, which consist of more than one agent, namely multi-agent systems. If a problem domain is quite complex, large, or unpredictable, then the only way it can reasonably be addressed is to develop a number of functionally specific and modular components (agents), which are specialised in solving a particular problem aspect. This decomposition allows each agent to use the most appropriate paradigm for solving its particular problems. When interdependent problems arise, the agents in the system must coordinate with one another to ensure that interdependencies are properly managed.

In the multi-agent system field, the key problem is the definition of an agent. There is still an ongoing debate, and little consensus, about the definition of an 'agent'. An increasing number of researchers and industrial practitioners have found that the following definition could be widely acceptable:

> "An agent is an encapsulated computational system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives." [3]

This definition implies that an agent should exhibit pro-active, reactive and social behaviour. Thus, the following key properties of an agent are required [4], [5].

1) Autonomy: agents are entities, which are clearly identifiable and problem solving. In addition, agents have well-defined boundaries and interfaces, which have control both over their internal states and over their own behaviour.
2) Reactivity: agents are situated (or embedded) in a particular environment. They receive inputs related to the states of their environment through sensor interfaces. Agents then respond in a timely fashion and act on the environment through effectors to satisfy their design objectives.
3) Pro-activeness: agents do not simply act in response to their environment. They are designed to fulfill specific purposes, namely that they have particular objectives (goals) to achieve. Agents are therefore able to exhibit goal-directed behaviour by taking the initiative and opportunistically adopting new goals.
4) Social Ability: agents are able to cooperate with humans and other agents in order to achieve their design objectives.

To intuitively understand what an agent is, it is worthwhile to consider some examples of agents [3].

- Any control system can be viewed as an agent. A simple example of such a system is a thermostat. A thermostat has a sensor for detecting room temperature. This sensor is directly embedded within the environment (i.e., the room), and it outputs one of two signals: one indicates that the temperature is too low and another indicates that the temperature is okay. The actions available to the thermostat are "heating on" or "heating off". The action "heating on" will generally have the effect of raising the room temperature. The decision making component of the thermostat implements (usually in electro-mechanical hardware) the following two rules: if the room temperature is too low, the action "heating on" is taken; if the room temperature is okay, the action "heating off" is taken.

This work was supported by two ARC Discovery Projects (DP150101775 and DP140100974) from the Australian Research Council, Australia.

D. Ye is with the School of Software and Electrical Engineering, Swinburne University of Technology, Australia, email: dye@swin.edu.au

M. Zhang is with the School of Computer Science and Software Engineering, University of Wollongong, Australia, email: minjie@uow.edu.au

Athanasios V. Vasilakos is with Lulea University of Technology, Sweden, email: vasilako@ath.forthnet.gr

- Most software daemons, (such as background processes in the UNIX operating system), which monitor a software environment and perform actions to modify it, can be viewed as agents. An example is the X Windows program xbiff. This program continually monitors a user's incoming emails and indicates via a GUI icon whether the user has unread messages. Whereas the thermostat agent in the previous example inhabits in a physical environment (the physical world), the xbiff program inhabits in a software environment. The xbiff program agent obtains information about this environment by carrying out software functions (e.g., by executing system programs), and the actions it performs are software actions (changing an icon on the screen or executing a program). The decision making component is just as simple as the thermostat example.

Multi-agent systems have been used in many industrial applications. The first multi-agent system applications appeared in the mid-1980s [1]. Up to now, multi-agent system applications have increasingly covered a variety of domains which range from manufacturing to process control [6], air-traffic control and information management [7].

B. *Overview of the multi-agent system design and development*

A multi-agent system is an extension of intelligent agent technology. In a multi-agent system, a group of autonomous agents act in an environment to achieve a common goal or their individual goals. These agents may cooperate or compete with each other and share or not share knowledge with each other [8], [9]. Since the concept of multi-agent systems is introduced, there have been several attempts to create methodologies to design and develop such systems [3], [10], [11], [12], [13], [14], [15], [16]. Development of a multi-agent system is difficult. A multi-agent system does not only have all the features of traditional distributed and concurrent systems, but also has exclusive difficulties due to the autonomy, flexibility and complex interactions of individual agents. As stated by Sycara [1], there is a lack of a proven methodology for designers to construct multi-agent systems for applications. Recently, Tran et al. [15] presented five stages in the multi-agent system development, which can summarise the basic development process of a multi-agent system.

1) Stage 1: goal analysis. This stage aims to understand the target problem domain and to specify the functionalities that the target multi-agent system should provide. The development should start with capturing system tasks, analysing the conflicts among these tasks and decomposing these tasks to small and easy-handled sub-tasks.
2) Stage 2: organisation design. In this stage, the organisational structure of the target multi-agent system is designed. Also, a set of agent classes which comprise the multi-agent system should be defined. The organisational structure can be constructed by defining a role for each agent class and specifying the authority relationships between these roles. The organisational structure refers to the application domain which the multi-agent system is developed to support, automate or monitor.
3) Stage 3: agent internal activity design. This stage focuses on the internal design of each agent class. The internal activities of each agent class include, for example, what goals an agent class is designed for, what knowledge this agent class has, when and how to respond to an internal or external event. An agent goal is a state of the world which an agent class is designed to achieve or satisfy [5]. The knowledge of an agent class is an agent belief which refers to the information that an agent hold about the world [17]. The response of an agent to events are agent plans which can be formed at run-time by planners or reasoners. An agent plan can be carried out based on some basic "if-then" rules which couple the states of the environment with the actions taken by agents.
4) Stage 4: agent interaction design. This stage defines the interactions between agent instances by designing a suitable interaction protocol or mechanism for the multi-agent system. The interaction protocol should specify the communication message format and how communication messages are transmitted, e.g., directly or indirectly. For the direct interaction mechanism, a suitable agent interaction protocol should be defined. This interaction protocol should be able to resolve any conflicts between agents and to ensure that all coordination rules governing the interaction are enforced. For the indirect interaction mechanism, the interaction protocol should be able to resolve conflicts not only between agents but also between agents and the tuple-center. Moreover, the interaction protocol should also model the tuple-center's behaviour.
5) Stage 5: architecture design. This stage concerns various implementation issues relating to the agent architecture and the multi-agent system architecture, e.g., selecting appropriate sensors to perceive the environment, selecting proper effectors to react to the environment and selecting a suitable implementation platform for implementing agents and the multi-agent system. The characteristics of the agents' perception, effect and communication should be specified at the design time. The internal constructs of each agent class, e.g., belief conceptualisation, agent goals and plans, should be mapped onto the architectural modules during implementation.

In this paper, the survey is delimited in Stage 2 and Stage 3, as most existing self-organisation mechanisms in multi-agent systems are developed in the two stages. The survey of other stages are left as one of our future studies.

C. *Self-organisation*

The term 'self-organisation' was introduced by Ashby in the 1960s [18], where self-organisation meant that some pattern was formed by the cooperative behaviour of individual entities without any external control or influence in a system. Phenomena of self-organisation can be found in natural biology. For example, there is no 'leader fish' in a school of fish but each individual fish has knowledge about its neighbours. Due to this localised and decentralised operation, the difficult task of forming and maintaining a scalable and highly adaptive shoal can be achieved [19], [20].

The ideas behind self-organisation have been widely used and studied in many fields, such as multi-agent systems [21], grid computing [22], sensor networks [23], [24], [25] and other industrial applications [26], [27]. Self-organisation has been proved to be an efficient way to deal with the dynamic requirements in distributed systems. Currently, there is still no commonly accepted exact definition of a self-organising system that holds across several scientific disciplines [20]. In multi-agent system field, Serugendo et al. [21] presented a definition of self-organisation.

> **Self-organisation** is defined as a mechanism or a process which enables a system to change its organisation without explicit command during its execution time [21].

Serugendo et al. further presented the definitions of strong self-organising systems and weak self-organising systems by distinguishing between systems where there is no internal and external explicit control from those where there is an internal centralised control (e.g., a termite society where the queen internally controls the behaviour of termites in the society).

> **Strong self-organising systems** are those systems where there is no explicit central control either internal or external.
>
> **Weak self-organising systems** are those systems where, from an internal point of view, re-organisation may be under an internal (central) control or planning.

In this paper, we consider only strong self-organising systems. Self-organisation has the following three properties [21], [28].

1) The absence of explicit external control. This property demonstrates that the system is autonomous. Adaptation and change of the system are based only on decisions of internal components without following any explicit external command. This property refers to the self- part of the above self-organisation definition.
2) Decentralised control. Self-organisation process can be achieved through local interactions among components without central control either internal or external. In addition, access to global information is also limited by the locality of interactions.
3) Dynamic and evolutionary operation. A self-organising system is able to evolve. When the environment changes, the self-organising system can evolve to adapt to the new environment and this evolution is independent of any external control. This property implies continuity of the self-organisation process.

Due to the above properties, self-organisation has been introduced into multi-agent systems for a long time to solve various problems in multi-agent systems [29]. Although many specific physical systems, such as multi-robot systems, sensor networks and so on, can be represented by multi-agent systems, multi-agent system itself is an independent research field and the research of multi-agent systems is independent of specific physical systems. The research of self-organisation in multi-agent systems mainly focuses on theoretical study while overlooking the requirements or constraints of specific physical systems. This is because researchers aim to design general self-organising multi-agent systems which could be applied in various physical systems (with proper modification if necessary). To the best of our knowledge, there is no survey of self-organisation mechanisms in general multi-agent systems, although surveys of self-organisation mechanisms in specific physical systems have been provided, e.g., the survey of self-organisation in cellular networks [30], the survey of self-organisation in ad hoc and sensor networks [31], [32], the survey of self-organisation for radio technologies [33], the survey of self-organisation in communications [34] and the survey of self-organisation in manufacturing control [35]. In this paper, a survey of self-organisation mechanisms in general multi-agent systems is provided. This survey classifies existing self-organisation mechanisms in general multi-agent systems, introduces their historical development, summarises and compares them, and points out future research directions. This survey is claimed as the contribution of this paper.

The rest of the paper is organised as follows. Section II presents related studies of the introduction or survey of self-organisation in multi-agent systems. Section III provides the classification of self-organisation mechanisms. Section IV surveys self-organisation mechanisms in multi-agent systems. Section V presents some applications of self-organising multi-agent systems. Section VI points out future research directions. Finally, Section VII concludes the paper.

II. Related Work

Although there is no survey of self-organisation mechanisms in multi-agent systems, some general introduction of self-organisation in multi-agent systems has been given. These general introduction articles make readers clearly understand what self-organisation is, the benefits of using self-organisation in multi-agent systems and the applications of self-organising multi-agent systems in real world systems.

In [21], Serugendo et al. concluded on a common definition of the concepts of self-organisation and emergence in multi-agent systems. They also summarised the properties and characteristics of self-organisation. Additionally, they developed an approach for selecting self-organisation mechanisms using a number of case studies and a set of evaluation criteria. Serugendo et al.'s work is the fundamental one which defines the concepts of self-organisation from a multi-agent system point of view.

In [28], Serugendo et al. further discussed the concepts of self-organisation and emergence. They then reviewed different classes of self-organisation mechanisms developed for use in various fields and studied the implementation of these mechanisms in multi-agent systems. The strengths and limits of these mechanisms were also examined. The self-organisation mechanisms reviewed in their paper, however, are not developed for use in multi-agent systems, while in this paper, the survey focuses on the self-organisation mechanisms developed to address various issues in multi-agent systems.

Tianfield and Unland [22] presented an overview on some interdisciplinary fields which have emerged in multi-agent systems and grid computing, e.g., semantic grids, autonomic computing and large-scale open multi-agent systems. They demonstrated that large-scale complex systems have a high desirability to be self-organising and they also reviewed existing studies which implemented self-organisation in those systems. However, only a small part of their review is on self-organisation in multi-agent systems, whereas in this paper, to give readers a clear understanding about the state of the art research of self-organisation in multi-agent systems, all of the surveyed studies are done in multi-agent systems.

Bernon et al. [36] described different mechanisms for generating self-organisation in multi-agent systems, which included self-organisation by reactive multi-agent systems, self-organisation using cooperative information agents, self-organisation by cooperation in adaptive multi-agent systems and self-organisation by holons. They then provided several examples of application of self-organising multi-agent systems to solve complex problems and discussed comparison criteria of self-organisation between different applications. However, as the work done in [22], only a small part of Bernon et al.'s work is on reviewing self-organisation mechanisms in multi-agent systems.

Picard et al. [37] studied how to make multi-agent organisations adapt to dynamics, openness and large-scale environments. Specifically, they compared two research views in detail, i.e., Agent-Centered Point of View (ACPV) and Organisation-Centered Point of View (OCPV) and studied how to apply these views to multi-agent systems. ACPV studies organisation from the point of view of emergent phenomena in complex systems, while OCPV focuses on designing the entire organisation and coordination patterns on the one hand, and the agents' local behaviour on the other hand. Their work, however, reviewed only the two views, i.e., ACPV and OCPV, and compared them but did not describe other self-organisation mechanisms.

In [38], Serugendo et al. generally discussed the existence of self-organisation in real-world systems, such as physical systems, biological systems, social systems, business and economic systems, and artificial systems. They then provided the applications of self-organisation in software systems, e.g., multi-agent systems, grid and P2P systems, network security, etc.. Their work is a general introduction of self-organisation and its applications, whereas this paper focuses on surveying self-organisation mechanisms in multi-agent systems.

Gorodetskii [29] analysed the state of the art in the field of multi-agent self-organising systems. Their work consists of three parts. The first part introduces the basic concepts of self-organisation and multi-agent systems. The second part presents the classification of self-organisation mechanisms in multi-agent systems. The remaining part provides examples of self-organisation mechanisms and their applications. All the examples of self-organisation mechanisms given by Gorodeskii are biology-based, e.g., swarm intelligence, nest building, web weaving, etc.. Compared to Gorodetskii's work, this paper will survey various self-organisation mechanisms developed for use in multi-agent systems.

According to the review of related work, it can be shown that current survey work of self-organisation mainly focuses on the concepts and the applications of self-organisation. Although some studies survey self-organisation along with multi-agent systems, they still mainly focus on the general introduction of self-organisation, multi-agent systems and their applications, while use only a small part to review specific self-organisation mechanisms. Also, some of these specific self-organisation mechanisms are not developed for use in multi-agent systems. In this paper, we intend to complement current related survey work by surveying existing self-organisation mechanisms which are developed to address specific issues in multi-agent systems.

III. Classification for Self-organisation

As stated in [29], currently, there is no conventional classification of the self-organisation mechanisms and different researchers use different features to make a classification. Based on the summarisation from [29], [30], generally, there are three classification methods for self-organisation mechanisms.

1) Objective-based classification. This classification focuses on the question of what the self-organisation mechanism is designed for. A self-organisation mechanism may be designed for task allocation, relation adaptation, etc.. Also, a self-organisation mechanism can be designed for multiple purposes, e.g., same self-organisation mechanism that aims for load-balancing can optimise capacity as well as quality of service.
2) Method-based classification. This classification focuses on the question of which method or technique is used to realise a self-organisation mechanism. A self-organisation mechanism may be designed based on reinforcement learning, where the driving force of the self-organisation process is a utility function and agents try to modify their behaviour so as to maximise their utility. A self-organisation mechanism may also be designed based on cooperation among agents, where self-organisation is achieved through local interactions between agents in a cooperative way.
3) Environment-based classification. This classification focuses on the question of which environment the self-organisation mechanism is designed in. A self-organisation mechanism may be designed in a multi-agent system, a sensor network or a grid system. Self-organisation mechanisms designed in different environments have to take specific requirements and constraints into account. If a self-organisation mechanism is designed in a wireless sensor network, due to the battery energy limitation of wireless sensors, interactions between sensors should be as few as possible, whereas such a constraint can be relaxed properly if the mechanism is designed in a general multi-agent system.

In this paper, we use the first classification method, objective-based classification, to classify self-organisation mechanisms. Because our survey is conducted in multi-agent system environments only, the third classification method, environment-based classification, cannot be used in this paper. If we use the second classification method, method-based classification, there will be a large number of technical contents. Technical contents, however, will harm the readability of this paper to some extent, especially for beginners. As this is a survey paper, good readability is the priority. Thus, the second classification method, method-based classification, is not very suitable in this paper. By using the first classification method, objective-based classification, readers can have a clear picture regarding not only the current important research issues in multi-agent systems, but also the advantages of using self-organisation to address these issues compared to those methods which do not use self-organisation.

IV. Survey of Self-organisation Mechanisms

As described in Section I, the development of a multi-agent system consists of five stages. In the last decade, the research of multi-agent systems has been thoroughly carried out in each stage. The five-stage development process is a top-down development process which begins from the requirement and goal analysis to the design of the conceptual architecture and the development of specific agent classes. Such development process, however, is infeasible in designing self-organising multi-agent systems, because (1) self-organising multi-agent systems are based on autonomous agents and their local interactions and (2) the global goal or behaviour cannot be specified or predicted in advance [29], [39]. Therefore, the development of self-organising multi-agent systems has to be carried out in a bottom-up way. Unfortunately, currently, there is a lack of a mature methodology or tool for developing self-organising multi-agent systems [39]. In this paper, the survey is conducted by reviewing the basic and important research issues in multi-agent systems so as to obey the bottom-up design process. According to objective-based classification, based on our investigation, there are six important research issues in multi-agent systems, which use self-organisation techniques. The six research issues are[1]:

1) task/resource allocation,
2) relation adaptation,
3) organisational design,
4) reinforcement learning,
5) enhancing software quality,
6) collective decision making.

Actually, the six research issues are overlapping to some extent. For example, reinforcement learning is often used as a tool to study other issues, e.g., task/resource allocation and relation adaptation. Also, task/resource allocation is often used as a platform for the study of other research issues, e.g., relation adaptation. Therefore, these research issues are not isolated but are closely related to each other. In this paper, the six research issues are selected for review, because (1) self-organisation techniques have been introduced into them and (2) they are the basic and important research issues in multi-agent systems. The two reasons make the review of them match the topic of this paper. There are some other important research issues in multi-agent systems, e.g., negotiation, coordination, planning and reasoning. However, because introducing self-organisation into these research issues has received little or no attention, these research issues are not reviewed in this paper. The discussion of these research issues about how to introduce self-organisation into them will be given in future research direction section (Section VI). Moreover, in order to demonstrate the historical development of these self-organisation mechanisms, we will also review a few representative non-self-organisation mechanisms, because the development of each self-organisation mechanism is usually based on previous non-self-organisation mechanisms. Researchers studied and summarised the limitations of non-self-organisation mechanisms and then, proposed self-organisation mechanisms to overcome these limitations.

A. Task/resource allocation

Task allocation and resource allocation are very important research issues in multi-agent systems, as many real-world problems can be modelled as task/resource allocation in multi-agent systems. Task allocation can be briefly described as that an agent has a task (or tasks) and cannot finish the task (or tasks) by itself, so the agent has to allocate the task (or tasks) to other agents to carry out. Then, how to efficiently and economically allocate tasks to other agents is the problem that task allocation mechanisms have to deal with. Resource allocation has a similar meaning to task allocation, where resource allocation focuses on how to efficiently allocate resources to agents so as to help them achieve their goals. In the following, we first review self-organising task allocation mechanisms and then self-organising resource allocation mechanisms.

Task allocation in multi-agent systems has been thoroughly studied and has a wide range of applications, e.g., target tracking in sensor networks [40] and labour division in robot systems [41]. Task allocation mechanisms in multi-agent systems can be classified into two categories: centralised and decentralised. Centralised task allocation mechanisms, e.g., [42] and [43], have the single point of failure and do not consider the change of tasks and agents. To overcome these drawbacks, decentralised task allocation mechanisms were developed, e.g., [44], [45], [46], [47], [48]. These decentralised mechanisms can avoid the single point of failure, but they still have some limitations. Scerri et al.'s approach [44] need a large amount of communication to remove conflicts, so it does not work well in large scale multi-agent systems. Abdallah and Lesser studied task allocation on the basis of game theory. Abdallah and Lesser's study considered only two agents and no discussion was given about how to extend their study to handle three or more agents. Weerdt et al. [46] used contract-net protocol [49], an auction based approach, for task allocation in multi-agent systems. In Weerdt et al.'s method, an agent allocates a task only to neighbours. Then, if an agent has few neighbours, its tasks may be difficult to allocate. Chapman et al.'s approach [47] is based on a distributed stochastic algorithm which is fast and needs few communication messages, but it may get stuck in local minima. Wang et al.'s mechanism [48] is based on ant colony algorithm which requires global pheromone matrix to achieve optimal solutions.

Self-organising task allocation mechanisms were also developed in multi-agent systems [50], [51]. The self-organising

[1]The two research issues, reinforcement learning and collective decision making, have been studied in both multi-agent system field and machine learning field. In this paper, we delimit the discussion in multi-agent systems, i.e., multi-agent learning and multi-agent collective decision making.

mechanisms are decentralised as well. Compared to centralised task allocation mechanisms, self-organising mechanisms can avoid the single point of failure. Compared to the non-self-organising decentralised task allocation mechanisms, self-organising mechanisms have good scalability and enable each agent to self-adapt its behaviour, without global information, for efficient task allocation in open and dynamic systems, where the set of tasks and agents may constantly change over time.

Macarthur et al. [50] proposed a distributed anytime algorithm for task allocation in open and dynamic multi-agent systems. Their algorithm is based on the fast-max-sum algorithm [52]. Macarthur et al. improved the fast-max-sum algorithm by presenting a pruning algorithm to reduce the number of potential solutions that need to be considered and by involving branch-and-bound search trees to reduce the execution time of fast-max-sum. Macarthur et al.'s algorithm is an online and anytime algorithm and it can self-adapt in dynamic environments. Thus, their algorithm has the self-organisation property, i.e., dynamically adapting itself without explicit control.

Santos et al. [51] proposed a swarm intelligence based clustering algorithm for task allocation in dynamic multi-agent systems. Their algorithm is inspired by the behaviour of forager bees, where a bee is considered an agent. During the clustering process, agents need to make a couple of decisions: whether to abandon an agent, whether to change to the group of the visited agent, whether to continue dancing to recruit other agents for a group, and whether to visit a dancer. The authors set a number of thresholds for agents to make decisions. In their algorithm, each agent can autonomously and dynamically make decisions based on current situations. Thus, their algorithm also has the self-organisation property.

Macarthur et al.'s algorithm [50] is based on the fast max-sum algorithm. Their algorithm is an anytime algorithm, so it can return a valid solution to a problem even if it is interrupted at any time before it ends. Also, the algorithm is expected to find better and better solutions, the more time it keeps running. Santos et al.'s algorithm [51] is based on bee colony which has a well-balanced exploration and exploitation ability.

Like task allocation, resource allocation in multi-agent systems has also been thoroughly studied and is relevant to a range of applications, e.g., network routing [53], manufacturing scheduling [54] and clouding computing [55], [56]. Resource allocation mechanisms can be either centralised or decentralised [57]. In centralised mechanisms, there is a single entity to decide on the allocation of resources among agents based on the constraints and preferences of each agent in the system. Typical examples for the centralised mechanism are combinatorial auctions [58], where the auctioneer is the central entity. In combinatorial auctions [59], [60], agents report their constraints and preferences to the auctioneer and the auctioneer makes the allocation of resources to the agents. The act of reporting constraints and preferences is called 'bidding'. An agent's bidding may be private or public to other agents based on requirements of the system. Bidding process may be operated in one round or multiple rounds. Based on the biddings, the auctioneer will make a decision on which resource is allocated to which agent. Typical decentralised mechanisms are usually operated through local interaction, such as the contract net approach [49] which consists of four interaction phases: announcement phase, bidding phase, assignment phase and confirmation phase. Many extensions to this protocol have been proposed. Sandholm [61] developed the TRACONET system which uses a variant of the contract net protocol to enable negotiation over the exchange of bundles of resources. Sandholm and Lesser [62] also extended the contract net protocol by enabling decommitment from agreed contracts during negotiation process with penalties applied, which gave agents more opportunities to find desirable partners. Aknine et al. [63] studied concurrent contract net protocol which allowed many managers negotiating simultaneously with many contractors. They added on the contract net protocol a pre-bidding phase and a pre-assignment phase, where agents proposed temporary bids and managers temporarily accepted or rejected these bids. In addition to negotiation, reinforcement learning is also an efficient approach for resource allocation. Schaerf et al. [64] proposed a resource allocation method based on reinforcement learning. In their method, when jobs arrive at agents, each agent independently decides on which resources are used to execute each job via reinforcement learning without interaction with other agents. Resources are dedicated to specific agents who do not make decisions during resource allocation. Only those agents, who have jobs to execute, make decisions. Tesauro [65] developed a similar model to Schaerf et al.'s work. There is a resource arbiter in Tesauro's model that dynamically decides resource allocation based on agents value functions which are learned independently. Zhang et al. [66] developed a multi-agent learning algorithm for online resource allocation in a network of clusters. In their algorithm, learning is distributed to each cluster, using local information only without accessing to the global system reward. The common limitation of these non-self-organising resource allocation mechanisms is that they are difficult to handle resource allocation in open and dynamic multi-agent systems. Therefore, resource allocation mechanisms, which have self-organisation properties, are also proposed.

Fatima and Wooldridge [67] presented an adaptive organisational policy, TRACE, for multi-agent systems. TRACE enables multi-agent organisations to dynamically and adaptively allocate tasks and resources between themselves to efficiently process an incoming stream of task requests. TRACE consists of two components: a task allocation protocol and a resource allocation protocol. The task allocation protocol, based on contract net [49], allows agents to cooperatively and efficiently allocate tasks to other agents which have the suitable capability and opportunity to carry these tasks out. The resource allocation protocol, based on computational market systems, enables resources to be adaptively and dynamically allocated to organisations to minimise the number of lost requests caused by an overload.

Schlegel and Kowalczyk [68] devised a distributed algorithm to solve the resource allocation problem in distributed multi-agent systems based on self-organisation of the resource consumers. In their algorithm, each resource consumer has several predictors to predict the resource consumption of each server, and uses this predictive result to allocate tasks to servers. Then, based on servers' feedback, each resource consumer evaluates the performance of its predictors and adjusts its predictors against each server.

An et al. [69] proposed an efficient negotiation method for resource allocation. In their method, negotiation agents can dynamically and autonomously adjust the number of tentative agreements for each resource and the amount of concession they are willing to make based on the situations of agents' vicinity environment. In addition, their method allows agents to decommit agreements by paying penalty and to dynamically modify the reserve price of each resource. Thus, agents have very high autonomy in their method.

Pitt et al. [70] complemented current principles of a resource allocation method by introducing the canons of distributive justice. The canons of distributive justice are represented as legitimate claims, which are implemented as voting functions that determine the order in which resource requests are satisfied. They then presented a formal model of a self-organising institution, where agents voted on the weight attached to the scoring functions. As a result, they unified principles of enduring self-organising institutions with canons of distributive justice to provide a basis for designing mechanisms to address the resource allocation problem in open systems.

Kash et al. [71] developed a dynamic model to fair divide resources between agents and they proposed desirable axiomatic properties for dynamic resource allocation mechanisms. They also designed two novel mechanisms which satisfied some of these properties. Their work is the first one which expands the scope of fair division theory from static settings to dynamic settings and which takes self-adaptation into account to fair division theory.

The work done in [68] and [69] aims at how to efficiently distribute resources to agents which make requests. Schlegel and Kowalczyk's method [68] is based on multi-agent learning, whereas An et al.'s method [69] is based on multi-agent negotiation. Negotiation techniques usually need more communication overhead than learning techniques and usually require more time to obtain a solution than learning techniques. However, negotiation techniques are more flexible and give agents more autonomy than learning techniques. The work done in [67] takes agents as resources and studies how to allocate and re-allocate agents to organisations in accordance with organisations' demands. Thus, the aim of [67] is different from that of [68] and [69]. The work done in [70] and [71] studies resource allocation using game theoretical approaches. The aim of [70] and [71] is to find an equilibrium that no agents have incentive to deviate from the allocation results. Thus, although all of these studies are about resource allocation, they focus on different aims and are suitable in different environments.

Summary: In self-organised task/resource allocation, there is no complex coordination mechanism among agents. Instead, the allocation process is a self-organised process that originates from local decisions made by each individual agent. Compared to non-self-organisation allocation mechanisms, e.g., [46], [48], [58], [63], self-organisation mechanisms [50], [51], [69], [70] are able to properly handle open and dynamic environments, where agents and tasks may be added or removed dynamically. In addition, self-organised allocation is robust to failures in communication and has good scalability in the number of agents. Specially, compared to multi-agent reinforcement learning allocation methods, e.g., [66], self-organisation mechanisms do not need a time-consuming convergence period. Table I summarises the characteristics of the aforementioned task/resource allocation approaches. In Table I, it can be seen that Macarthur et al.'s self-organising task allocation approach is based on the fast-max-sum algorithm [50], while Santos et al.'s approach is based on swarm intelligence [51]. Both of the two approaches are decentralised and have good scalability. The fast-max-sum algorithm can exploit a particular formulation of task allocation environments to greatly reduce the communication message size and computation required when applying max-sum in dynamic environments. In [51], swarm intelligence is used to form agent groups for task allocation given that an individual agent does not have enough resources to complete a task. In the swarm intelligence based approach, agents use only local information and follow simple rules to derive intelligent global behaviour. Thus, such approach is very suitable in the environments, where each individual agent has only incomplete information about the environments. For self-organising resource allocation approaches, auction based approach [67] and negotiation based approach [69] can achieve optimal results, because results are obtained through the bargaining of both parties, which is unlike other approaches [68], [70], [71] that derive results by using only a specific algorithm or a specific set of algorithms. However, during the bargaining process, heavy communication overhead cannot be avoided. Thus, such auction and negotiation based approaches are not suitable in some environments where communication resources are intensive, e.g., wireless sensor networks.

B. *Relation adaptation*

The term, relation adaptation, in different fields has different meanings. In web-based systems, relation adaptation means extracting new types of relations that exist between entities in a system [78]. Here, in multi-agent systems, relation adaptation, also known as relation modification, is a subfield of self-organisation, which studies how to modify relations between agents to achieve an efficient agent network structure. A relation adaptation mechanism enables agents to arrange and rearrange the structure of the multi-agent system in order to adapt to changing requirements and environmental conditions [76]. As relation adaptation is a subfield of self-organisation rather than a field which is independent of self-organisation, there are no 'non-self-organising' relation adaptation mechanisms. Therefore, here, we directly review the work done on relation adaptation in multi-agent systems.

Gaston and desJardins [72] developed two network structural adaptation strategies for dynamic team formation. Their first strategy was a structure-based approach, where an agent prefers to select another agent to form a connection, which has

TABLE I
CHARACTERISTICS OF THE TASK/RESOURCE ALLOCATION APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Macarthur et al. [50]	max-sum algorithm	decentralised	local	good	can get a solution at any time	memory intensive
Santos et al. [51]	swarm intelligence	decentralised	local	good	well-balanced exploration and exploitation	slow computation speed
Fatima and Wooldridge [67]	auction and market-based modeling	decentralised	both	neutral	optimal results can be obtained	large communication overhead
Schlegel and Kowalczyk [68]	reinforcement learning	decentralised	local	good	little communication overhead	large computation overhead
An et al. [69]	negotiation	decentralised	local	neutral	optimal results can be obtained	large communication overhead
Pitt et al. [70]	voting	decentralised	local	good	precise equilibrium can be obtained	large communication overhead
Kash et al. [71]	fair division theory	centralised	global	neutral	precise equilibrium can be obtained	a central controller and global information are required

TABLE II
CHARACTERISTICS OF THE RELATION ADAPTATION APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Gaston and desJardins [72]	reasoning	decentralised	local	good	easy to implement	only one relation type is considered
Glinton et al. [73]	reasoning	decentralised	local	good	easy to implement	only one relation type is considered
Abdallah and Lesser [74]	reinforcement learning	decentralised	local	good	fast and efficient	large computation overhead only one relation type is considered
Griffiths and Luck [75]	tag-based modeling	decentralised	local	good	suitable in dynamic environments	large communication overhead only one relation type is considered
Kota et al. [76]	reasoning	decentralised	local	good	two relation types are considered	computation results are biased from one agent towards another
Ye et al. [77]	trust modeling and reinforcement learning	decentralised	local	good	two relation types are considered, relation weights are considered	large computation overhead

more neighbours. Their second strategy was a performance-based approach, where an agent prefers to form a connection with the agent who has better performance. The two strategies are suitable in different situations.

Glinton et al. [73] analysed the drawback of the structure-based strategy proposed in [72] empirically, and then designed a new network adaptation strategy to limit the maximum number of links that an agent could have.

Abdallah and Lesser [74] did further research into relation adaptation of agent networks and creatively used reinforcement learning to adapt the network structure. Their method enables agents not only to adapt the underlying network structure during the learning process but also to use information from learning to guide the adaptation process.

Griffiths and Luck [75] presented a tag-based mechanism for supporting cooperation in the presence of cheaters by enabling individual agents to change their neighbourhoods with other agents. Griffiths and Luck's mechanism is very suitable in particular dynamic environments where trust or reputation among agents is difficult to establish.

Kota et al. [76] devised a relation adaptation mechanism. Their work is the first one, which takes multiple relations and relation management cost into account. The relation adaptation algorithm in their mechanism is based on meta-reasoning and enables agents to take the actions, which can maximise their utilities at each step.

Ye et al. [77] proposed a composite relation adaptation mechanism. Their mechanism consists of three elements. The first one is a trust model to enable agents to use not only their own experience but also other agents' opinions to select candidates, which can make agents select the most valuable candidates to adapt relations. The second one is a multi-agent Q-learning algorithm to enable two agents to independently evaluate their rewards about adapting relations and to balance exploitation and exploration. The third one is the introduction of weighted relations into the relation adaptation mechanism. The introduction of weighted relations can improve the performance of the mechanism and make the mechanism more suitable in dynamic environments.

Summary: The work done in [72], [73], [74], [75] assumed that only one type of relation existed in the network and the number of neighbours possessed by an agent had no effect on its local load. These assumptions are impractical in some cases where multiple relations exist among agents in a network and agents have to expend resources to manage their relations with other agents. Kota et al.'s work [76] took multiple relations and relation management load into account. All of these studies, however, considered only crisp relations between agents and oversimplified candidate selection, while Ye et al.'s work [77] considered weighted relations, where there is a relation strength, ranged in $[0, 1]$, to indicate how strong the relation is between two agents, and employed a trust model to select candidates to adapt relations. However, as Ye et al.'s work is based on both trust modeling and reinforcement learning, the computation overhead is large. Moreover, during the trust building process, agents have to communicate with one another, so the communication overhead is also heavy. Table II summarises the characteristics of the aforementioned relation adaptation approaches.

C. Organisational design

The research of organisational self-design can be traced back to 1977. Weick [79] discussed the application of the concept of self-designing systems in social organisations. At that time, the concept of self-design was so new that concrete illustrations of it in business organisations were rare. However, the benefits of self-design were revealed then. In the face of swift changes in the environment, organisations will do too little, too late and will even fail. Also, organisations have to avoid having someone from the outside come in to rewire the organisations, whereas organisations have to do the rewiring themselves. Therefore, self-design becomes the only choice of organisations.

Organisational design generally refers to how members of a society act and relate with one another [80]. It can be used to design and manage participants' interactions in multi-agent systems. Specifically, organisational design includes assigning agents different roles, responsibilities and peers, and also assigning the coordination between the roles and the number of resources to the individual agents. Different designs applied to the same problem will have different performance characteristics. Thus, it is important to understand the features of different designs. Organisational self-design has been introduced, which allows agents to self-design, i.e., self-assign roles, responsibilities and peers between agents. Like relation adaptation, organisational self-design is also a sub-field of self-organisation in multi-agent systems, so there are no 'non-self-organising' organisational self-design mechanisms. Here, we directly review the work done on organisational self-design in multi-agent systems.

Decker et al. [81] developed a multi-agent system, in which agents can adapt at organisational, planning, scheduling and execution levels. Specifically, their work focused on agent cloning for execution-time adaptation towards load-balancing, when an agent recognises, via self-reflection, that it is becoming overloaded.

Shehory et al. [82] proposed an agent cloning mechanism, which subsumed task transfer and agent mobility. To perform cloning, an agent has to reason about its current and future loads and its host's load, as well as the capabilities and loads of other machines and agents. Then, the agent may decide to create a clone or transfer tasks to other agents or migrate to another host. Their work discusses in detail when and how an agent makes a clone for task allocation in a distributed multi-agent environment.

In [83], Ishida et al. studied organisational self-design as an adaptive approach to work allocation and load-balancing. Their approach allows two agents to combine into one agent if these two agents are idle and allows one agent to divide into two agents if that agent is overloaded. However, their approach does not consider agent self-extinction. Moreover, their approach is designed only for a specific problem: work-allocation and load-balancing in distributed production systems.

Kamboj and Decker [84] extended Ishida et al.'s work by including worth-oriented domains, modelling other resources in addition to processor resources and incorporating robustness into the organisational structures. Later, Kamboj [85] analysed the tradeoffs between cloning and spawning in the context of organisational self-design and found that combining both cloning and spawning could generate more suitable organisations than using those mechanisms, which use only a single approach.

In [86], Ye et al. provided an organisational self-design mechanism which enables agents to clone and spawn new agents, and these cloned and spawned agents can merge in future if necessary. For an individual agent, spawning is triggered when it cannot finish the assigned tasks on time. If a task or several tasks in its list cannot be completed before the expiry time, an agent will spawn one or several apprentice agent(s), each of which has a corresponding resource to complete a task. Cloning happens when an agent has too many neighbours, which means that the agent has a heavy overhead for managing relations with other agents. Spawned agents will be self-extinct if no more tasks have to be carried out, and cloned agents merge with original agents if the number of neighbours decreases.

Summary: The work done in [81] and [83] focused on specific systems, i.e., a financial portfolio management system and a distributed production system, respectively, so they may not be suitable for other systems. Shehory et al.'s work [82] overlooks agent merging and self-extinction and this overlook may yield a large number of redundant and idle agents. Kamboj's work [84], [85] is under a particular computational framework, TAEMS [87] (Task Analysis, Environment Modeling and Simulation), where tasks are represented using extended hierarchical task structures. This binding may limit the usability of their approaches in other domains. Ye et al.'s work [86] does not focus on specific systems and is not under an existing computational framework. In addition, Ye et al's work takes agent merging and self-extinction into consideration. Thus, it can overcome the limitation of the aforementioned work to some extend. The application of such cloning mechanisms, however, is limited in physical systems, e.g., robot systems and sensor networks, as the components in these physical systems are hardware and cannot be cloned. Table III summarises the characteristics of the aforementioned organisational design approaches.

D. Reinforcement learning

Reinforcement learning is the problem faced by an agent that must learn behaviour through trial-and-error interactions with a dynamic environment [92], [93]. At each step, the agent perceives the state of the environment and takes an action which causes the environment to transit into a new state. The agent then receives a reward signal that evaluates the quality of

TABLE III
CHARACTERISTICS OF THE ORGANISATIONAL DESIGN APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Decker et al. [81]	reasoning	decentralised	local	good	little communication and computation overhead	focus on a specific system
Shehory et al. [82]	reasoning	decentralised	local	good	little communication and computation overhead	overlook agent merging and self-extinction
Ishida et al. [83]	reasoning	decentralised	local	good	little communication and computation overhead	focus on a specific system
Kamboj and Decker [84], [85]	reasoning	decentralised	local	good	little communication and computation overhead	established on a specific computational framework
Ye et al. [86]	reasoning	decentralised	local	good	little communication and computation overhead	large computation overhead

TABLE IV
CHARACTERISTICS OF THE REINFORCEMENT LEARNING APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Kiselev and Alhajj [88], [89]	hierarchical clustering	decentralised	local	good	optimal results can be obtained	large communication overhead
Zhang et al. [90], [91]	market-based modeling	hybrid	local	good	optimal results can be obtained	large communication overhead

this transition. As stated in [93], there are two main strategies for solving reinforcement-learning problems. The first strategy is to search in the space of behaviours in order to find one that performs well in the environment. This approach has been taken by work in genetic algorithms and genetic programming [94]. The second strategy is to use statistical techniques and dynamic programming methods to estimate the utility of taking actions in states of the world. The research of reinforcement learning in multi-agent systems mainly focuses on the second strategy, because the second strategy takes advantage of the special structure of reinforcement learning problems that is not available in optimisation problems in general [93]. Reinforcement learning in multi-agent systems has three new challenges [95]. First, it is difficult to define a good learning goal for the multiple reinforcement learning agents. Most of the times, each learning agent must keep track of the other learning agents. Only when the agent is able to coordinate its behaviour with other agents' behaviour, a coherent joint behaviour can be achieved. Second, as the learning process of agents is non-stationary, the convergence properties of most reinforcement learning algorithms are difficult to obtain. Third, the scalability of algorithms to realistic problem sizes is a great concern, as most multi-agent reinforcement algorithms focus on only two agents. There are a large number of multi-agent learning algorithms developed in the last decade, e.g., [96], [97]. There are also some outstanding survey papers of multi-agent reinforcement learning, e.g., [93], [95]. As the aim of this paper is to survey self-organisation instead of learning, we just review some latest and representative multi-agent learning algorithms in this subsection. Zhang and Lesser [91] proposed a gradient-based learning algorithm that augmented a basic gradient ascent algorithm with policy prediction. Their algorithm removes some strong assumptions from existing algorithms, e.g., [98], [99], [100] by enabling agents to predict others' policies. Thus, their algorithm has better scalability and is more suitable to real applications compared to existing algorithms. Later, Zhang and Lesser [33] developed a learning approach that generalised previous coordinated DCOP (Distributed Constraint OPtimisation) based multi-agent reinforcement learning approaches, e.g., [101], [102], [103], which needed intensive computation and significant communication among agents. In comparison, Zhang and Lesser's approach [33] enable multi-agent reinforcement learning to be conducted over a spectrum from independent learning without communication by enabling agents to compute their beneficial coordination set in different situations. Elidrisi et al. [104] proposed a fast adaptive learning algorithm for repeated stochastic games. Compared to some existing algorithms which require a large number of interactions among agents, their algorithm requires only a limited number of interactions. Elidrisi et al. achieved their goal by developing 1) a meta-game model to abstract a stochastic game into a lossy matrix game representation, 2) a prediction model to predict the opponents' next action and 3) a reasoning model to reason about the next action to play given the abstracted game and the predicted opponents actions. Piliouras et al. [97] proposed an analytical framework for multi-agent learning. Unlike standard learning approaches, Piliouras et al.'s work does not focus on the convergence of an algorithm to an equilibrium or on payoff guarantees for the agents. Instead, they focus on developing abstractions that can capture the details about the possible agents' behaviours of multi-agent systems, in which there are rich spatiotemperal correlations amongst agents' behaviours.

The introduction of self-organisation into reinforcement learning is presented recently[2], which aims to improve the learning performance by enabling agents to dynamically and autonomously adjust their behaviours to suit changing environments.

Kiselev and Alhajj [88], [89] proposed a computationally

[2]Please do not confuse this 'self-organisation' with 'self-organising map' which is a type of artificial neural network. Self-organisation mentioned in this paper is a notion which can benefit other fields by enabling agents to autonomously make decisions and dynamically adapt to environmental changes.

efficient market-based self-adaptive multi-agent approach to continuous online learning of streaming data and provided a fast dynamic response with event-driven incremental improvement of optimisation results. Based on the self-adaptive approach, the performance of the continuous online learning is improved and the continuous online learning can adapt to environmental variations. The approach is based on an asynchronous message-passing method of continuous agglomerative hierarchical clustering and a knowledge-based self-organising multi-agent system for implementation.

Zhang et al. [90] integrating organisational control into multi-agent reinforcement learning to improve the learning speed, quality and likelihood of convergence. Then, they introduced self-organisation into organisational control to further enhance the performance and reduce the complexity of multi-agent reinforcement learning [91]. Their self-organisation approach groups strongly interacting learning agents together, whose exploration strategies are coordinated by one supervisor. The supervisor of a group can buy/sell agents from/to other groups through negotiation with the supervisors of those groups.

Summary: Multi-agent reinforcement learning is an efficient and scalable method to solve many real world problems, e.g., network packet routing and peer-to-peer information retrieval. However, due to factors including a non-stationary learning environment, partial observability, a large number of agents and communication delay between agents, reinforcement learning may converge slowly, converge to inferior equilibria or even diverge in realistic environments [91]. Self-organisation then can be used to organise and coordinate the behaviours of agents based on their current states of learning. Thus, self-organisation can not only improve the quality of agent learning but also can make agents efficiently learning in dynamic environments. Table IV summarises the characteristics of the aforementioned reinforcement learning approaches. The work of Kiselev and Alhajj [89] focuses on a specific problem: continuous online clustering of streaming data, while the work of Zhang et al. [91] focuses on a common problem in multi-agent learning: convergence. Although the two studies have different focuses, both of them use the multi-agent negotiation technique to realise the self-organisation process. As both of their proposed approaches are based on multi-agent negotiation, both of them suffer the large communication overhead.

E. Enhancing software quality

Current software systems have ultra large scales due to the explosion of information and complexity of technologies. Software systems, thus, require new and innovative approaches for building, running and managing so as to become more versatile, flexible, robust and self-optimising by adapting to changing operational environments or system characteristics [111]. Agent-based software engineering has also been studied for a long time [4], [112], which is concerned with how to effectively engineer agent systems, that is, how to specify, design, implement, verify (including testing and debugging), and maintain agent systems [113]. Strictly speaking, agent-based software engineering is not a traditional research issue in multi-agent systems, instead it is an application of agent technology. However, the study of agent-based software engineering is significant for developing and implementing multi-agent systems. Thus, we still review the studies of self-organisation for agent-based software engineering in this paper.

To enhance agent-based software quality, several techniques have been proposed, e.g., agile techniques [114] and data mining techniques [115]. Agile techniques can handle unstable requirements throughout the development life cycle and can deliver products in shorter time frames and under budget constraints in comparison with traditional development methods. Data mining techniques can be used to discover and predict faults and errors in software systems. Self-organisation can also be used in agent-based software systems to enhance software quality. Compared to those techniques, self-organisation technique enables agents to self-diagnose faults and errors in software systems. Thus, self-organiation technique has good scalability and can be used in large scale agent-based software systems. Self-orgainsing agent-based software systems, which are able to adjust their behaviours in response to the perception of the environment, have become an important research topic. Cheng et al. [111] and Lemos et al. [116] provided a research roadmap regarding the state of the art research progress and the research challenges of developing, deploying and managing self-adaptive software systems. Based on their summary, there are four essential topics of self-adaptive software systems: design space for self-adaptive solutions, software engineering processes for self-adaptive solutions, decentralisation of control loops, and practical run-time verification and validation.

Georgiadis et al. [105] studied the feasibility of using architectural constraints as the basis for the specification, design and implementation of self-organising architectures for distributed software systems. They developed a fully decentralised runtime system to support structural self-organisation based Darwin component model [117] and showed that the required architectural styles can be expressed and subsequently analysed in a simple set based logical formalism.

Malek et al. [106] presented a self-adaptive solution for the redeployment of a software system to increase the availability of the system. Their solution is based on a collaborative auctioning algorithm, where the auctioned items are software components. Each host is represented as an autonomous agent and agents sell and buy software components between them through the auctioning algorithm. By redeploying software components, both the availability and robustness of the software system can be increased.

Iftikhar and Weyns [107] proposed a formalised architecture model of a self-adaptive software system and used model checking to verify behavioural properties of the software system. They also proved a number of self-adaptation properties for flexibility and robustness based on a case study, i.e., a decentralised traffic monitoring system. The traffic monitoring software system is conceived as an agent-based system consisting of two components, *agent* and *organisation middleware*. The *agent* is responsible for monitoring the traffic

TABLE V
CHARACTERISTICS OF THE SOFTWARE QUALITY ENHANCEMENT APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Georgiadis et al. [105]	architectural constraints	decentralised	local	good	simplifying the analysis of software systems	low computation speed
Malek et al. [106]	collaborative auction	decentralised	local	good	increasing robustness of software systems	large communication overhead
Iftikhar and Weyns [107]	model checking	decentralised	local	good	guaranteeing flexibility and robustness of software systems	large computation overhead
Iglesia and Weyns [108], [109]	hierarchical model and formal verification	hybrid	local	good	guaranteeing the requirements of software systems being met	large communication and computation overhead

TABLE VI
CHARACTERISTICS OF THE COLLECTIVE DECISION MAKING APPROACHES

Papers	Techniques used	Centralised/ Decentralised	Local/Global information required	Scalability	Pros	Cons
Valentini et al. [110]	voting	decentralised	local	good	fast and efficient	large communication and computation overhead

and collaborating with other agents to report a possible traffic jam to clients. The *organisation middleware* offers life cycle management services to set up and maintain organisations.

Iglesia and Weyns [108], [109] introduced a self-adaptive multi-agent system which is an architectural approach that integrates the functionalities provided by a multi-agent system with software qualities offered by a self-adaptive solution. They then presented a reference model for the self-adaptive multi-agent system and applied it to a mobile learning case. They also used a formal verification technique as an approach to guarantee the requirements of the self-adaptive multi-agent system application. The reference model is a three-layered architecture where the bottom layer provides the communication infrastructure which defines the means for communication between agents, the middle layer provides the multi-agent system which handles requirements of the domain, and the top layer provides self-adaptation which can modify the multi-agent system layer to cover system quality concerns.

Summary: Self-adaptation is a well-known approach for managing the complexity of modern software systems by separating logic that deals with particular runtime qualities [118], [119]. Self-adaptation enables a software system to adapt itself autonomously to internal dynamics and changing conditions in the environment to achieve particular quality goals. Self-adaption in software systems includes a number of self-* properties, e.g., self-healing, self-protection and self-optimisation, to address changing operating conditions in the system. For example, self-healing enables a software system to automatically discover, diagnose and correct faults; self-protection enables a software system to autonomously prevent from both internal and external malicious attacks; self-optimisation enables a software system to monitor and adapt resource usage to ensure optimal functioning relative to defined requirements. Overall, self-adaptation is a promising approach for modern software systems. Table V summarises the characteristics of the aforementioned software quality enhancement approaches. The similarity of these studies is that all of them aim to enhance software quality, while the difference of them is that they focus on different aspects of agent-based software engineering. Georgiadis et al. [105] focused on how to build a self-organising architecture as a basis for distributed software systems development. Georgiadis et al.'s arthitecture can work well in the environment, where components may suddenly fail without the opportunity to interact with the rest of the system. Their architecture, however, cannot handle dynamic environments, where events may dynamically rebind and system requirements may dynamically change. Malek et al. [106] focused on how to increase the availability of a system. Malek et al.'s method is decentralised and does not need global knowledge of system properties. Thus, the method can scale to the exponentially complex nature of the redeployment problem. However, their method is based on an auction algorithm, so the communication overhead of their method is heavy. Iftikhar and Weyns [107] focused on how to check and verify the self-adaptation properties of a self-organising software system. Iftikhar and Weyns's model can enhance the validation of software system qualities by transferring formalization results over different phases of the software life cycle. However, their model is proposed and evaluated through a case study, i.e., a traffic monitoring system. Thus, it is unclear how their model works in other systems. Iglesia and Weyns [108], [109] focused on how to design a general model to cover various concerns of system quality. By using behavioural models and formal methods, Iglesia and Weyns's approach can guarantee the correctness of system behaviour and guarantee the quality properties of interest during the engineering of self-organising multi-agent systems. However, the implemented system, based on their approach, has not been evaluated in dynamic environments. Thus, it is unclear if the desired quality goals of the system can be met in undesired states.

F. Collective decision making

Collective decision making originates from social science. When a person is in a social context, her decisions are influenced by those of others. Then, collective decision making is a process where the members of a group decide on a course of action by consensus [120]. Collective decision making has been studied by economists and sociologists since at least the 1970s [121], [122]. Later, collective decision making has been studied by statistical physicists who developed models to quantitatively describe social and economic phenomena that involve large numbers of interacting people [123], [124]. Recently, collective decision making has also been investigated in multi-agent systems [110]. Traditional solutions of collective decision making are centralised [125]. Self-organisation can provide a valuable alternative to the centralised solutions. However, introducing self-organisation into collective decision making is a significant challenge because only local perception and local communication can be used [110]. Globally defined consensus time and decision accuracy are both difficult to predict and guarantee. Towards this end, several self-organised collective decision making algorithms have been proposed [126], [120], [127], [110]. Among these algorithms, only one was developed in multi-agent systems [110].

Valentini et al. [110] presented a weighted voter model to implement a self-organised collective decision making process to solve the best-of-n decision problem in multi-agent systems. They also provided an ODE (ordinary differential equations) model and a master equation model to investigate the system behaviour in the thermodynamic limit and to investigate finite-size effects due to random fluctuations. The weighted voter model is based on the extension of classic voter model by (1) considering the change of agents' neighbourhood, (2) allowing agents to participate in the decision process at different rounds for a time proportional to the qualities of their opinions and (3) allowing agents to temporarily leave the decision pool in order to survey the quality of their current opinion. Valentini et al. used opinion-based approaches. Opinion-based approaches need more communication overhead than the swarm intelligence technique. However, a consensus is easier and faster to achieve using opinion-based approaches than using swarm intelligence technique. The advantages of their approach includes (1) with the increase of system size, the decision accuracy also increases, (2) with the increase of system size, the consensus time logarithmically increases and (3) the approach is robust to noisy assessments of site qualities. However, as the approach is opinion-based, the generation and transmission of opinions in the system are computation and communication intensive.

Summary: The introduction of self-organisation into collective decision making makes the decision making process decentralised and enables agents to dynamically make decisions based on environmental changes. However, as described above, collective decision making in self-organised systems is still challenging because it relies only on local perception and local communication. Table VI summarises the characteristics of the work in [110].

G. Other research issues

In addition to the above issues, there are some other issues in multi-agent systems which are also addressed using self-organisation mechanisms.

1) Coalition formation: In some real systems, e.g., distributed sensor networks, individual agents often need to form coalitions to accomplish complex tasks, as complex tasks cannot be performed by a single agent or groups may perform more efficiently with respect to the single agents' performance. Most existing coalition formation studies enable each individual agent to join only one coalition (see [128] for a survey of existing coalition formation mechanisms). To overcome this limitation, some researchers proposed overlapping coalition formation [129] and fuzzy coalition formation [130] to enable each agent to join multiple coalitions. Such studies, however, do not allow agents to dynamically adjust degrees of involvement in different coalitions. Ye et al. [128] introduced self-adaptation into coalition formation by allowing agents to dynamically adjust degrees of involvement in different coalitions and to join new coalitions. Through the introduction of self-adaptation, the performance of the coalition formation mechanism is improved in terms of agents' profit and time consumption. Ye et al.'s approach, however, is based on negotiation, so it suffers from a large communication overhead.

2) Evolution of cooperation: The evolution of cooperation among selfish individuals is a fundamental issue in a number of disciplines, such as artificial intelligence [131], [132], physics [133], biology [134], sociology [135] and economics [136]. The aim of evolution of cooperation is to increase the proportion of cooperators in a group of agents, each of which is either a cooperator or a defector. Existing strategies of the evolution of cooperation have both strengths and limitations. For example, some strategies can only increase the proportion of cooperators, only if the initial proportion of cooperators is larger than a specific number, e.g., 0.5; some strategies can only increase the proportion of cooperators, only if it works in a specific network structure, e.g., a small-world network [137]. Ye and Zhang [138] developed a self-adaptation based strategy for evolution of cooperation by embodying existing strategies as each agent's knowledge and letting each agent dynamically select a strategy to update its action, i.e., cooperate or defect, according to different environmental situations. As a result, Ye and Zhang's strategy can utilise the strengths of existing strategies and avoid the limitations of them. Ye and Zhang's strategy, however, is based on a reinforcement learning algorithm. Thus, its performance is highly dependent on the performance of the learning algorithm. Such dependency relationship limits the applicability of their strategy, as a learning algorithm is suitable only in a limited number of situations.

3) Self-checking logical agents: Certification and assurance of agent systems constitute crucial issues, as agents represent a particularly complex case of dynamic, adaptive and reactive software systems [139]. Certification is aimed at producing evidence indicating that deploying a given system in a given context involves the lowest possible level of risk of adverse consequences. Assurance is related to dependability, i.e., to ensuring that system users can rely on the system. Costantini and Gasperis [140], [141], [139] have done a lot of work on self-checking agent systems. They presented a comprehensive framework for runtime self-monitoring and self-checking assurance of logical agents by means of temporal-logic-based special constraints to be dynamically checked at a certain (customisable) frequency. The constraints are based on a simple interval temporal logic, A-ILTL (agent-oriented interval linear temporal logic). Based on Costantini and Gasperis's framework, agent systems are able to dynamically self-check the violations of desired system properties. Moreover, in the case of violation, agents can quickly restore to a desired state by means of run-time self-repair. However, their framework mainly focuses on self-checking and self-repair while overlooks other self-* functionality of agent systems, e.g., self-healing, self-optimisation, etc..

V. Applications of self-organising multi-agent systems

In addition to theoretical studies, self-organising multi-agent systems can be used in many application domains [36], [39]. In this section, some examples of application of self-organising multi-agent systems are provided.

In [142], [143], George et al. developed a self-organising multi-agent system for flood forecasting. The system consists of several stations installed over the river basin which forecast local variation in the water level. Each station has a two-level multi-agent architecture. The lower level includes sensors which detect variations of water level every hour and provide the data to upper level agents. Each upper level agent then makes its forecast based on the data, provided by sensors, and the assessment of the quality of its preceding forecasts. The self-organisation process is carried out at the level of sensors, where each sensor dynamically modifies the weights of measurements taken at different times. Experiments demonstrated that the proposed self-organising multi-agent system is applicable to the actual evolution of the water level even at the early stage of the system operation when only a small number of learning samples have been used.

Camurri et al. [144], [145] proposed a self-organising multi-agent system for controlling road traffic in a large city. Traffic participants, i.e., cars, are represented by car software agents. Traffic lights are represented by light agents. The aim of the system is to coordinate individual cars and to control traffic lights so as to minimise traffic jams. In the system, car software agents are coordinated using the information obtained from light agents. The basic idea of the self-organisation paradigm is that car software agents dynamically select routes for cars to avoid current traffic jams based on the information obtained from light agents. Meanwhile, light agents implement a context-sensitive traffic lights control strategy to minimise traffic jams throughout the city.

In [146], [147], a self-organising multi-agent system was proposed to control manufacturing resources. The self-organisation mechanism is based on a swarm intelligence model which controls the production processes by predicting the resource utilisation for a short period of time, evaluating the state of order execution and finding the best further routing for the orders. The self-organising multi-agent system includes three types of agents: a product agent, an order agent and a resource agent. The three types of agents indirectly coordinate to find variants of the step-by-step order execution using concrete resources and to generate an optimal product execution plan.

Dury et al. [148] described an application of a self-organising multi-agent system in land utilisation. The system is used to optimally assign farming territories to various crops so as to obtain the maximum total profit by selling the crops yielded in the future. In this assignment problem, the resource to be assigned is the set of farming lots with characteristics such as area, soil type, distance to the nearest villages and transportation infrastructure. The self-organising multi-agent system involves a set of agents which compete for capturing the lots. Agents are self-organised into groups. The agents in the same group want to get hold of the lots for the same crop. Each agent of each group competes for capturing a lot with the desired properties. If an agent wins it makes a contribution to the utility function of its group.

Shorabi et al. [149] presented three protocols/algorithms for self-organisation of wireless sensor networks, where each sensor is represented as an agent. The first protocol is the self-organising medium access control protocol which is for network start-up and link-layer organisation of wireless sensor networks. The first protocol is used to form a flat topology for wireless sensor networks. The first protocol is a distributed one which enables sensor nodes to discover their neighbours and establish transmission/reception schedules for communicating with them without the need for local or global master nodes. The second algorithm is the eavesdrop-and-register algorithm which is used for seamless interconnection and mobility management of mobile nodes in wireless sensor networks. The third protocol consists of three algorithms: 1) the sequential assignment routing algorithm which is for multi-hop routing, 2) the single winner election algorithm and 3) the multi-winner election algorithm which handle the necessary signaling and data transfer tasks in local cooperative information processing.

In multi-robot systems, self-organisation can be used for the division of labour control [150], [151], [152]. For example, Liu et al. [151] presented a self-adaptation mechanism dynamically adjust the ratio of foragers to resters in a swarm of foraging robots in order to maximise the net energy income to the swarm. The self-adaptation mechanism is based only on local sensing and communications. By using this mechanism, robots can use internal information (e.g., successful food retrieval), environmental information (e.g., collisions with team mates while searching for food) and social information (e.g., team mate success in food retrieval) to dynamically vary the time spent on foraging and resting.

VI. Future research directions

The technology of self-organising multi-agent systems integrates the properties of self-organisation, e.g., decentralisation and dynamic and evolutionary operation, and the advantages of multi-agent systems, e.g., autonomy and sociability. Self-organising multi-agent systems, therefore, have good scalability, are robust to failures of components and can adapt to the dynamics of external environments and the changing of internal structures. The majority of current research of self-organising multi-agent systems is theoretical. The study on application of self-organising multi-agent systems is still in an early stage. Thus, as a whole, the future study of self-organising multi-agent systems should focus more on real world systems by considering specific constraints and requirements. Moreover, since currently there is a lack of a mature methodology or tool for developing self-organising multi-agent systems, the future research could also focus on devising an efficient methodology or tool for developing self-organising multi-agent systems[3].

To design and develop self-organising multi-agent systems, various self-organisation mechanisms have to be developed to address the basic research issues described in Section IV. These research issues are important not only in multi-agent systems but also in specific physical systems, e.g., task allocation in multi-robot systems, resource allocation in sensor networks, etc.. Although a number of self-organisation mechanisms in multi-agent systems have been developed in the last decades, there is still room for them to be improved or extended. In this section, the future research directions of self-organisation mechanisms against each important research issue in multi-agent systems and some specific physical systems will be discussed.

A. Task/Resource allocation

Task allocation and resource allocation are traditional and important research issues in multi-agent systems. Existing self-organising task/resource allocation approaches in multi-agent systems mainly focus on how to efficiently allocate tasks and resources. Most of these approaches, however, overlook how to re-allocate tasks and resources if such re-allocation can bring more benefits to the focal agents. Thus, the future research on self-organising task/resource allocation can be extended to self-organising task/resource allocation and re-allocation. Such re-allocation could be based on the performance of existing agents and the capability of new agents. In addition, most of existing self-organising approaches do not consider the interdependencies among tasks and resources [155], [156]. Thus, the future research can take such interdependencies into account. Also, most of existing self-organising approaches were developed in selfish environments, where every agent aims to maximise its own benefit. However, many real world systems, e.g., sensor networks, are cooperative environments, where agents aim to maximise the overall benefit of a system. Thus, it is also important to develop self-organising task/resource allocation approaches in cooperative environments.

Self-organising task allocation in general multi-agent systems, however, has not attracted much attention. This is because task allocation is often used as a platform for other research issues, e.g., coalition formation [42] and relation adaptation [76]. Also, task allocation is often studied incorporating with specific physical systems, e.g., sensor networks [157] and multi-robot systems [151]. Thus, in this situation, the future research of self-organising task allocation should concentrate on specific physical systems by taking specific constraints and requirements into account. Such physical systems include sensor networks, multi-robot systems, grid computing, manufacturing control and so on, where self-organisation is highly desirable to increase the autonomy, scalability and robustness of these physical systems. Likewise, self-organising resource allocation could also concentrate on specific physical systems, e.g., sensor networks [158] and smart grid [159]. These physical systems may be open and dynamic and are difficult to manage or organise using existing non-self-organisation techniques. However, it should be noted that in different physical systems, resources may have different properties. For example, resources may be continuous or discrete. In a smart grid, resource (i.e., energy) is continuous while in a fruit market, resource (i.e., fruit) is discrete. Resources may be re-usable or not. In a computer system, resource (e.g., CPU or memory) is re-usable while in a smart grid, resource (i.e., energy) is not re-usable. All such properties must be taken into consideration when designing self-organising resource allocation mechanisms in specific physical systems.

B. Relation adaptation

Relation adaptation is actually researched as a subfield of self-organisation. In [74], [160], [77], the authors used the terms ‘relation adaptation’ and ‘self-organisation’ interchangeably. Relation adaption did not attract much attention in multi-agent systems compared to those popular research issues, e.g., task/resource allocation, coalition formation, reinforcement learning. The research on relation adaptation usually has to deal with two problems: with whom to modify relations and how to modify relations. The first problem is about selecting partners in a network and can usually be addressed using trust models. Existing approaches for partner selection, however, are based on agents interaction history but do not consider the dynamism of the environment. For example, if agent i and agent j has a good interaction history. Then, based on existing approaches, i will add j as one of its neighbours. However, due to the dynamism of the environment, j may leave the environment very shortly. Thus, in this situation, if i takes environmental dynamism into account, it will not add j as one of its neighbours. The second problem is about selecting a proper relation, in the case that the number of relations is more than one, and can usually be addressed using reasoning and learning (either individual or collective depending on the settings). However, agents which use reasoning and learning techniques will become very subjective, as there is no interaction between agents during reasoning and learning processes. Thus, the negotiation technique may be a good choice in this situation, because both parties can present their requirements and offers. Therefore, a result which can benefit both parties can be achieved.

[3]Although there have been some methodologies for developing self-organising multi-agent systems [153], [143], [154], [36], they are far from mature.

In addition, relation adaptation can be used in real-world systems, e.g., social networks [161] and multi-robot systems [162], to adapt relations among entities to achieve more efficient organisational structures. Therefore, it is also a future research direction to study relation adaptation in real-world systems.

C. Organisational design

Organisational design was initially studied in social organisations for specific purposes. Then, the research was conveyed in multi-agent systems for general purposes. Originally, the research of organisational design focuses on how to assign roles to different participants in an organisation. When self-organisation is introduced in organisational design, i.e., organisational self-design, the research includes other aspects, e.g., agent cloning/spawning, agent extinction and agent mobility. However, there lacks an organisational self-design framework which combines all these aspects together: self-assigning roles, self-cloning, self-spawning, self-extinction, etc.. Therefore, future research of organisational self-design can be conducted through this way.

Also, as the research of organisational design was originated from social organisations, in the future, the research of organisational self-design can be conducted in social organisations, e.g., enterprises [163]. Compared to existing organisational techniques in social organisatiosn, introducing orgainsational self-design technique into social organisations can increase the autonomy of each entity and avoids the centralisation of authority to some extent.

D. Reinforcement learning

Like task/resource allocation, reinforcement learning is also an important and a popular research topic in multi-agent systems. However, introducing self-organisation into reinforcement learning has not attracted much attention. The milestone work in this field is Zhang et al.'s work [90], [91], which introduced organisational control and self-organisation into reinforcement learning. The self-organisation approach used in Zhang et al's work carries out in the management layer, i.e., between supervisors of groups. Future research may focus on designing a self-organisation approach which is able to work not only in the management layer but also between agents in each group. In addition, the essence of reinforcement learning is how to adjust probability distribution among available actions and many adjustment approaches have been proposed. Thus, another future research may be that existing probability distribution adjustment approaches can be embodied as knowledge of each agent and each agent autonomously selects an adjustment approach in each learning round. Also, in reinforcement learning, the setting of values of learning parameters, e.g., learning rates, can affect the performance of a learning algorithm, and no set of values of parameters is best across all domains [164]. However, in most existing learning algorithms, the setting of values of parameters are hand-tuned. Thus, it should be interesting to develop a self-organisation approach to self-adjust the values of learning parameters during the learning process in different situations.

Reinforcement learning has been employed in many physical systems. For example, reinforcement learning can be used for sleep/wake-up scheduling in wireless sensor networks to save sensors' energy [165], [166]. Reinforcement learning can also be used to learn the pricing strategies for the broker agents in smart grid markets [167]. Therefore, future research can also focus on applying self-organising reinforcement learning in physical systems to improve the learning performance in these systems, e.g., increasing convergence speed or reducing communication and computation overhead.

E. Enhancing software quality

Traditionally, software quality is guaranteed by system managers. However, an error happening in software systems may cost system managers several hours, sometimes even several days, to find and fix it. Therefore, self-organisation is introduced into software systems, which includes many self-* properties, e.g., self-configuration, self-checking, self-healing, self-protection, self-optimisation etc.. Most existing studies include part of these self-* properties. Thus, future research may design a self-organising agent-based software system which includes all of these properties. This is certainly a very large project and would need a group of researchers to collaboratively complete.

F. Collective decision making

Like organisational design, collective design making also originates from social science, where members of a group have to collectively make decisions to achieve a consensus. Most existing studies of self-organising collective decision making were conducted in multi-robot systems instead of general multi-agent systems. Very recently, Valentini et al. [110] investigated self-organisation for collective decision making in multi-agent systems. Their work considered nearly every aspect of self-organising collective decision making. Future research of self-organising collective decision making may be conducted in open environments where new agents can join the group and existing agents can leave the group. In addition, as the problem of finding a collective agreement over the most favorable choice among a set of alternatives is the 'best-of-n' decision problem, in a dynamic environment, the 'n' may change over time. Hence, it is necessary to develop a self-adaptive approach to enable agents to self-adjust their behaviour in a timely manner to make a best decision in the dynamic environment.

Self-organising collective decision making can also be applied in sensor networks for various purposes, e.g., clock synchronisation. Most of existing techniques for clock synchronisation in sensor networks need global information or require all sensors to participate in the synchronisation process [168], [169]. By using self-organising collective decision making technique for clock synchronisation, only local information is needed and only part of sensors are required to participate.

G. Other research issues

In addition to the above research issues, there are some other important research issues in multi-agent systems, which attracted little or no attention on how to introduce self-organisation into them. These research issues include coalition formation, evolution of cooperation, self-checking logical agents, negotiation, coordination, planning and reasoning.

Coalition formation and evolution of cooperation have been studied for a very long time. However, very few studies considered introducing self-organisation into them. Thus, the research of introducing self-organisation into coalition formation and evolution of cooperation is very potential. As both coalition formation and evolution of cooperation consist of a number of steps, future research of introducing self-organisation into them can focus on different steps. For example, for coalition formation, an existing study [128] uses self-organisation for agents to self-adapt degrees of involvement in different coalitions, while future research could use self-organisation for agents to autonomously and dynamically recruit/expel coalition members. For evolution of cooperation, an existing study [86] uses self-organisation for agents to autonomously select an action update strategy in each round, while future research could use self-organisation for agents to modify the relationships (e.g., strengthen or weaken the relationships) with their neighbours in each round.

Most of the research on self-checking logical agents has been undertaken by Costantini and Gasperis [140], [141], [139]. The research on self-checking logical agents is akin to the research on self-checking software agents except that Costantini and Gasperis considered more on logical agents. Therefore, similar to the issue, enhancing software quality, future research on self-checking logical agents could extend to other self-* properties of logical agents, e.g., self-healing, self-optimisation, etc..

In multi-agent systems, negotiation is a key tool for multiple autonomous agents to reach mutually beneficial agreements. The process of negotiation can be of different forms, e.g., auctions, protocols and bargains. In each of the forms, there is a set of rules which govern the interaction process among agents. Such rules indicate the allowable participants (e.g., which agents are allowed to join the negotiation), the negotiation states (e.g., bids or offers generated, accepted or rejected, negotiation started and negotiation terminated), the events that cause state transitions (e.g., when a bid or an offer is accepted, the negotiation is terminated; or when the deadline is reached, the negotiation is terminated no matter if an agreement is achieved) and the valid actions of the participants in particular states (e.g., which can be sent by whom, to whom and at when) [170]. In the future, self-organisation may be introduced into negotiation for agent decision making in the interaction process. For example, agents may self-adjust the strategies for bid or offer generation and dynamically decide when to generate bids or offers in different situations based on self-organisation mechanisms.

In order to successfully interact in environments, agents must be able to reason about their interactions with other heterogeneous agents which have different properties and capabilities [171]. During the reasoning process, an agent first observes the environment and its internal state. Then, the agent creates a new goal and generates a set of candidate plans. Finally, the agent selects the most suitable plan to execute to achieve the goal [172]. The plan generation is called planning. Multi-agent planning is also known as multi-agent sequential decision making, that is a set of agents with complementary capabilities coordinate to generate efficient plans so as to achieve their respective goals [173], [174], [175]. These plans should not be in conflict with each other. Reasoning, planning and coordination have a close relationship with one another, as planning is a step during a reasoning process and coordination is used to guarantee that individual agents' plans are not in conflict with each other. In the future, self-organisation may be introduced into planning and coordination. For example, self-organisation mechanisms can be developed for adaptive generation and selection of plans. Also, as coordination can be carried out using learning [173], [176], [175], self-organisation in coordination may be achieved by designing self-organising learning algorithms such as the ones discussed in Section IV.

Moreover, there are delay phenomena, e.g., time delay and feedback delay, in practical self-organising systems, e.g., biological systems [177], [178], neural network systems [179], [180]. These delay phenomena, however, have not been considered in existing self-organising multi-agent systems, although delay phenomena have been taken into account in general multi-agent systems [181]. In order to make self-organising multi-agent systems applicable in practical systems, it is also the future research to take delay phenomena into account when designing self-organising multi-agent systems.

VII. Conclusion

In this paper, self-organisation mechanisms in multi-agent systems have been surveyed. The classification method used in this survey is objective-based classification in order to provide good readability. Readers then can have a deep understanding of the benefits of using self-organisation to address various mult-agent system research issues. In this survey, we have provided the basic concepts of self-organisation, have highlighted the major research issues in multi-agent systems, have discussed how these issues can be addressed using self-organisation approaches, and have presented important research results achieved. We have also identified other survey papers regarding self-organisation in multi-agent systems and pointed out the differences between their work and ours. Finally, the paper is concluded with a discussion of future research directions of those surveyed research issues in multi-agent systems. The research issues discussed in this paper have been broadly studied not only in multi-agent systems but also in other specific systems, e.g., robot systems and sensor networks. Thus, each of the research issues deserves a separate survey, which is one of our future studies. Moreover, as described in Section I, the survey in this paper delimits in Stage 2: organisation design and Stage 3: agent internal activity design. Thus, in the future, the survey could be extended to other stages of multi-agent system development.

VIII. Acknowledgments

This work was originally created at the School of Computer Science and Software Engineering, University of Wollongong.

References

[1] K. P. Sycara, "Multiagent systems," *AI Magazine*, vol. 19, pp. 79–92, 1998.

[2] H. Simon, *Models of Man: Social and Rational - Mathematical Essays on Rational Human Behavior in a Social Setting*. New York: Wiley, 1957.

[3] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[4] M. Wooldridge, "Agent-based software engineering," *IEE Proceedings on Software Engineering*, pp. 26–37, 1997.

[5] M. J. Wooldridge, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, G. Weiss, Ed. MIT Press, 1999.

[6] B. Chaib-draa, "Industrial applications of distributed ai," *Communications of the ACM*, vol. 38, no. 11, pp. 49–53, 1995.

[7] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *JAAMAS*, vol. 1, pp. 7–38, 1998.

[8] M. Wooldridge, *An Introduction to MultiAgent Systems, 2nd Edition*. Wiley, 2009.

[9] P. G. Balaji and D. Srinivasan, *Innovations in Multi-Agent Systems and Applications - 1*. Springer, 2010, ch. An Introduction to Multi-Agent Systems, pp. 1–27.

[10] D. Kinny, M. Georgeff, and A. Rao, "A methodology and modelling technique for systems of bdi agents," in *Proc. of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, vol. 1038, 1996, pp. 56–71.

[11] H. S. Nwana, "Software agents: An overview," *Knowledge Engineering Review*, vol. 11, no. 3, pp. 205–244, 1996.

[12] C. Iglesia, M. Garijo, and J. Gonzalez, "A survey of agent-oriented methodologies," *Lecture Notes in Computer Science*, vol. 1555, pp. 185–198, 1998.

[13] M. Wooldridge, N. Jennings, and D. Kinny, "The gaia methodology for agent-oriented analysis and design," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 285–312, 2000.

[14] M. F. Wood and S. A. DeLoach, "An overview of the multiagent systems engineering methodology," in *Proc. of the First International Workshop on Agent-Oriented Software Engineering*, vol. 1957, 2001, pp. 207–221.

[15] Q. N. Tran and G. Low, "Mobmas: A methodology for ontology-based multi-agent systems development," *Information and Software Technology*, vol. 50, pp. 697–722, 2008.

[16] O. Z. Akbari, "A survey of agent-oriented software engineering paradigm: Towards its industrial acceptance," *Journal of Computer Engineering Research*, vol. 1, no. 2, pp. 14–28, 2010.

[17] Y. Shoham and S. B. Cousins, *Foundations of Knowledge Representation and Reasoning*. Springer, 1994, ch. Logics of mental attitudes in AI, pp. 296–309.

[18] W. Ashby, *Principles of the self-organizing system*. Pergamon Press, 1962, ch. Principles of self-organization: Transactions of the University of Illinois Symposium, pp. 255–278.

[19] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. Annual Conf. on Computer Graphics and Interactive Techniques SIGGRAPH 87*, 1987, pp. 25–34.

[20] W. Elmenreich, R. D'Souza, C. Bettstetter, and H. deMeer, "A survey of models and design methods for self-organizing networked systems," *Self-Organizing Systems*, vol. 5918, pp. 37–49, 2009.

[21] G. D. M. Serugendo, M. Gleizes, and A. Karageorgos, "Self-organization in multi-agent systems," *The Knowledge Engineering Review*, vol. 20, pp. 165–189, 2005.

[22] H. Tianfield and R. Unland, "Towards self-organization in multi-agent systems and grid computing," *Multiagent and Grid Systems*, vol. 1, pp. 89–95, 2005.

[23] A. Rogers, E. David, and N. R. Jennings, "Self-organized routing for wireless microsensor networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, pp. 349–359, 2005.

[24] Y. Song, L. Liu, H. Ma, and A. V. Vasilakos, "A biology-based algorithm to minimal exposure problem of wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 11, pp. 417–430, 2014.

[25] L. Liu, Y. Song, H. Zhang, H. Ma, and A. V. Vasilakos, "Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks," *IEEE Transactions on Computers*, 2014, dOI10.1109/TC.2013.229, 2014.

[26] R. Frei and G. D. M. Serugendo, "Self-organizing assembly systems," *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 41, pp. 885–897, 2011.

[27] M. A. Khan, H. Tembine, and A. V. Vasilakos, "Game dynamics and cost of learning in heterogeneous 4g networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 198–213, 2012.

[28] G. D. M. Serugendo, M. P. Gleizes, and A. Karageorgos, "Self-organisation and emergence in mas: An overview," *Informatica*, vol. 30, pp. 45–54, 2006.

[29] V. I. Gorodetskii, "Self-organization and multiagent systems: I. models of multiagent self-organization," *Journal of Computer and Systems Sciences*, vol. 51, pp. 256–281, 2012.

[30] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys and Tutorials*, vol. 15, pp. 336–361, 2013.

[31] K. L. Mills, "A brief survey of self-organization in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 7, pp. 823–834, 2007.

[32] F. Dressler, "A study of self-organisation mechanisms in ad hoc and sensor networks," *Computer Communications*, vol. 31, pp. 3018–3029, 2008.

[33] Z. Zhang, K. Long, and J. Wang, "Self-organization paradigms and optimization approaches for cognitive radio technologies: A survey," *IEEE Wireless Communications*, pp. 36–42, 2013.

[34] S. D. et. al., "A survey of autonomic communications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, pp. 223–259, 2006.

[35] S. Bousbia and D. Trentesaux, "Self-organization in distributed manufacturing control: State-of-the-art and future trends," in *Proc. of 2002 IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[36] C. Bernon, V. Chevrier, V. Hilaire, and P. Marrow, "Applications of self-organising multi-agent systems: An initial framework for comparison," *Informatica*, vol. 30, pp. 73–82, 2006.

[37] G. Picard, J. F. Hubner, O. Boissier, and M.-P. Gleizes, "Reorganisation and self-organisation in multi-agent systems," in *Proc. of the International Workshop on Organizational Modeling'09*, 2009.

[38] G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos, *Self-organising Systems*. Springer, 2011, ch. Self-organisation and Emergence in Software Systems, pp. 7–32.

[39] V. I. Gorodetskii, "Self-organization and multiagent systems: Ii. applications and the development technology," *Journal of Computer and Systems Sciences*, vol. 51, pp. 391–409, 2012.

[40] M. Sims, C. V. Goldman, and V. Lesser, "Self-organization through bottom-up coalition formation," in *Proc. of AAMAS'03*, 2003, pp. 867–874.

[41] P. R. Ferreira, F. S. Boffo, and A. L. C. Bazzan, "A self-organized algorithm for distributed task allocation in complex scenarios," in *Proc. of Workshop on Coordination and Control in Massively Multi-Agent Systems (CCMMS'07)*, 2007, pp. 19–33.

[42] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, pp. 165–200, 1998.

[43] X. Zheng and S. Koenig, "Reaction function for task allocation to cooperative agents," in *Proc. of AAMAS'08*, 2008.

[44] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe, "Allocating tasks in extreme teams," in *Proc. of AAMAS'05*, 2005, pp. 727–734.

[45] S. Abdallah and V. Lesser, "Learning the task allocation game," in *Proc. of AAMAS'06*, 2006, pp. 850–857.

[46] M. d. Weerdt, Y. Zhang, and T. Klos, "Distributed task allocation in social networks," in *Proc. of AAMAS'07*, 2007, pp. 500–507.

[47] A. Chapman, R. A. Micillo, R. Kota, and N. R. Jennings, "Decentralised dynamic task allocation: A practical game-theoretic approach," in *Proc. of AAMAS'09*, 2009.

[48] L. Wang, Z. Wang, S. Hu, and L. Liu, "Ant colony optimization for task allocation in multi-agent systems," *Communications Software*, vol. 10, pp. 125–132, 2013.

[49] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. 29, pp. 1104–1113, 1980.

[50] K. S. Macarthur, R. Stranders, S. D. Ramchurn, and N. R. Jennings, "A distributed anytime algorithm for dynamic task allocation in multi-agent systems," in *Proc. of AAAI'11*, 2011.

[51] D. S. Santos and A. L. C. Bazzan, "Distributed clustering for group formation and task allocation in multi-agent systems: A swarm intelligence," *Applied Soft Computing*, vol. 12, pp. 2123–2131, 2012.

[52] S. D. Ramchurn, , A. Farinelli, K. S. Macarthur, and N. R. Jennings, "Decentralised coordination in robocuprescue," *The Computer Journal*, vol. 53, pp. 1447–1461, 2010.

[53] R. Feldmann, M. Gairing, T. Lucking, B. Monien, and M. Rode, "Selfish routing in non-cooperative networks: a survey," in *Proc. of 28th International Symposium on Mathematical Foundations of Computer Science*, 2003, pp. 21–45.

[54] P. Sousa, C. Ramos, and J. Neves, "The fabricare scheduling prototype suite: Agent interaction and knowledge base," *Journal of Intelligent Manufacturing*, vol. 14, pp. 441–455, 2003.

[55] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing*, vol. 54, pp. 252–269, 2010.

[56] M. R. Rahimi, N. Venkatasubramanian, and A. V. Vasilakos, "Music: Mobility-aware optimal service allocation in mobile cloud computing," in *Proc. of IEEE CLOUD'13*, 2013, pp. 75–82.

[57] Y. Chevaleyre and et al., "Issues in multiagent resource allocation," *Informatica*, vol. 30, pp. 3–31, 2006.

[58] P. Cramton, Y. Shosham, and R. Steinberg, *Combinatorial Auctions*. MIT Press, 2006.

[59] V. Krishna, *Auction Theory*. Academic Press, 2002.

[60] P. R. Wurman, M. P. Wellman, and W. E. Walsh, "A parametrization of the auction design space," *Games and Economic Behaviour*, vol. 35, pp. 304–338, 2001.

[61] T. W. Sandholm, "An implementation of the contract net protocol based on marginal cost calculations," in *Proc. of AAAI'93*, 1993, pp. 256–262.

[62] T. W. Sandholm and V. R. Lesser, "Leveled commitment contracts and strategic breach," *Games and Economic Behaviour*, vol. 35, pp. 212–270, 2001.

[63] S. Aknine, S. Pinson, and M. F. Shakun, "An extended multi-agent negotiation protocol," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 8, pp. 5–45, 2004.

[64] A. Schaerf, Y. Shoham, and M. Tennenholtz, "Adaptive load balancing: A study in multi-agent learning," *Journal of Artificial Intelligence Research*, vol. 2, pp. 475–500, 1995.

[65] G. Tesauro, "Online resource allocation using decompositional reinforcement learning," in *Proc. of AAAI'05*, 2005, pp. 886–891.

[66] C. Zhang, V. Lesser, and P. Shenoy, "A multi-agent learning approach to online distributed resource allocation," in *Proc. of IJCAI'09*, 2009, pp. 361–366.

[67] S. S. Fatima and M. Wooldridge, "Adaptive task and resource allocation in multi-agent systems," in *Proc. of AGENTS'01*, 2001.

[68] T. Schlegel and R. Kowalczyk, "Towards self-organising agent-based resource allocation in a multi-server environment," in *Proc. of AAMAS'07*, 2007, pp. 74–81.

[69] B. An, V. Lesser, and K. M. Sim, "Strategic agents for multi-resource negotiation," *Autonomous Agents and Multi-Agent Systems*, vol. 23, pp. 114–153, 2011.

[70] J. Pitt, J. Schaumeier, D. Busquets, and S. Macbeth, "Self-organising common-pool resource allocation and canons of distributive justice," in *Proc. of IEEE Sixth International Conference on Self-Adaptive and Self-Organising Systems*, 2012.

[71] I. Kash, A. D. Procaccia, and N. Shah, "No agent left behind: Dynamic fair division of multiple resources," in *Proc. of AAMAS'13*, 2013, pp. 351–358.

[72] M. E. Gaston and M. desJardins, "Agent-organized networks for dynamic team formation," in *Proc. of AAMAS'05*, Utrecht, Netherlands, Jul. 2005, pp. 230–237.

[73] R. Glinton, K. Sycara, and P. Scerri, "Agent organized networks redux," in *Proc. of AAAI'08*, Chicago, 2008, pp. 83–88.

[74] S. Abdallah and V. Lesser, "Multiagent reinforcement learning and self-organization in a network of agents," in *Proc. of AAMAS'07*, Honolulu, Hawai'i, May 2007, pp. 172–179.

[75] N. Griffiths and M. Luck, "Changing neighbours: Improving tag-based cooperation," in *AAMAS'10*, Toronto, Canada, May 2010, pp. 249–256.

[76] R. Kota, N. Gibbins, and N. R. Jennings, "Decentralized approaches for self-adaptation in agent organizations," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, pp. 1:1–28, 2012.

[77] D. Ye, M. Zhang, and D. Sutanto, "Self-organization in an agent network: A mechanism and a potential application," *Decision Support System*, vol. 53, pp. 406–417, 2012.

[78] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Relation adaptation: Learning to extract novel relations with minimum supervision," in *Proc. of IJCAI'11*, 2011, pp. 2205–2210.

[79] K. E. Weick, "Organization design: Organizations as self-designing systems," *Organizational Dynamics*, vol. 6, pp. 30–45, 1977.

[80] B. Horling and V. Lesser, "Using quantitative models to search for appropriate organizational designs," *Autonomous Agents and Multi-Agent Systems*, vol. 16, pp. 95–149, 2008.

[81] K. Decker, K. Sycara, and M. Williamson, "Cloning for intelligent adaptive information agents," in *Revised Papers from the Second Australian Workshop on Distributed Artificial Intelligence*, 1996, pp. 63–75.

[82] O. Shehory, K. Sycara, P. Chalasani, and S. Jha, "Agent cloning: An approach to agent mobility and resource allocation," *IEEE Communications Magazine*, vol. 36, pp. 58–67, 1998.

[83] T. Ishida, L. Gasser, and M. Yokoo, "Organization self-design of distributed production systems," *IEEE Trans. on Knowl. Data Engin.*, vol. 4, no. 2, pp. 123–134, Apr. 1992.

[84] S. Kamboj and K. S. Decker, "Organizational self-design in semi-dynamic environments," in *AAMAS'07*, Honolulu, Hawai'i, USA, May 2007, pp. 1228–1235.

[85] S. Kamboj, "Analyzing the tradeoffs between breakup and cloning in the context of organizational self-design," in *AAMAS'09*, Budapest, Hungary, May 2009, pp. 829–836.

[86] D. Ye, M. Zhang, and D. Sutanto, "Cloning, resource exchange and relation adaptation: An integrative self-organisation mechanism in a distributed agent network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 887–897, 2014.

[87] V. Lesser, K. Decker, and T. Wagner, "Evolution of the gpgp/taems domain-independent coordination framework," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 9, no. 1, pp. 87–143, 2004.

[88] I. Kiselev and R. Alhajj, "A self-organizing multi-agent system for adaptive continuous unsupervised learning in complex uncertain environments," in *Proc. of AAAI'08*, 2008, pp. 1808–1809.

[89] ——, "An adaptive multi-agent system for continuous learning of streaming data," in *Proc. of WI-IAT'08*, 2008, pp. 148–153.

[90] C. Zhang, S. Adballah, and V. Lesser, "Integrating organizational control into multi-agent learning," in *Proc. of AAMAS'09*, 2009, pp. 757–764.

[91] C. Zhang, V. Lesser, and S. Adballah, "Self-organization for coordinating decentralized reinforcement learning," in *Proc. of AAMAS'10*, 2010, pp. 739–746.

[92] G. I. P. Athanasios V. Vasilakos, "A new approach to the design of reinforcement schemes for learning automata: Stochastic estimator learning algorithm," *Neurocomputing*, vol. 7, pp. 275–297, 1995.

[93] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[94] J. Schmidhuber, *Evolutionary Computation: Theory and Applications*, 1996, ch. A General Method for Multi-Agent Learning and Increamental Self-Improvement in Unrestricted Environment.

[95] L. Busoniu, R. Babuska, and B. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man and Cybernetics: Part C - Applications and Reviews*, vol. 38, pp. 156–172, 2008.

[96] C. Zhang and V. Lesser, "Coordinating multi-agent reinforcement learning with limited communication," in *Proc. of AAMAS'13*, 2013, pp. 1101–1108.

[97] G. Piliouras, C. Nieto-Granda, H. I. Christensen, and J. S. Shamma, "Persistent patterns: Multi-agent learning beyond equilibrium and utility," in *Proc. of AAMAS'14*, 2014, pp. 181–188.

[98] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, pp. 215–250, 2002.

[99] B. Banerjee and J. Peng, "Generalized multiagent learning with performance bound," *Autonomous Agents and Multi-Agent Systems*, vol. 15, pp. 281–312, 2007.

[100] S. Abdallah and V. Lesser, "A multiagent reinforcement learning algorithm with non-linear dynamics," *Journal of Artificial Intelligence Research*, vol. 33, pp. 521–549, 2008.

[101] C. Guestrin, M. G. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in *Proc. of ICML'02*, 2002, pp. 227–234.

[102] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, pp. 1789–1828, 2006.

[103] C. Zhang and V. R. Lesser, "Coordinated multi-agent reinforcement learning in networked distributed pomdps," in *Proc. of AAAI*, 2011.

[104] M. Elidrisi, N. Johnson, M. Gini, and J. Crandall, "Fast adaptive learning in repeated stochastic games by game abstraction," in *Proc. of AAMAS'14*, 2014, pp. 1141–1148.

[105] I. Georgiadis, J. Magee, and J. Kramer, "Self-organising software architectures for distributed systems," in *Proc. of WOSS'02*, 2002.

[106] S. Malek, M. M. Rakic, and N. Medvidovic, *Component Deployment*. Springer, 2005, ch. A Decentralized Redeployment Algorithm for Improving the Availability of Distributed Systems, pp. 99–114.

[107] M. U. Iftikhar and D. Weyns, "A case study on formal verification of self-adaptive behaviors in a decentralised system," in *Proc. of the 11th International Workshop on Foundations of Coordination Languages and Self Adaptation*, 2012.

[108] D. G. Iglesia and D. Weyns, "Enhancing software qualities in multi-agent systems using self-adaptation," in *Proc. of European Workshop on Multi-Agent Systems*, 2012, pp. 1–15.

[109] ——, "Sa-mas: Self-adaptation to enhance software qualities in multi-agent systems," in *Proc. of AAMAS'13*, 2013, pp. 1159–1160.

[110] G. Valentini, H. Hamann, and M. Dorigo, "Self-organized collective decision making: The weighted voter model," in *Proc. of AAMAS'14*, 2014, pp. 45–52.

[111] B. H. C. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, "Software engineering for self-adaptive systems: A research roadmap," *Self-Adaptive Systems*, vol. LNCS 5525, pp. 1–26, 2009.

[112] N. R. Jennings, "On agent-based software engineering," *Artificial Intelligence*, vol. 117, pp. 277–296, 2000.

[113] M. Winikoff, "Future directions for agent-based software engineering," *International Journal of Agent-Oriented Software Engineering*, vol. 3, pp. 402–410, 2009.

[114] A. Hossain, M. A. Kashem, and S. Sultana, "Enhancing software quality using agile techniques," *Journal of Computer Engineering*, vol. 10, pp. 87–93, 2013.

[115] Z. Liaghat, A. H. Rasekh, and A. R. Tabebordbar, "Enhance software quality using data mining algorithms," in *Proc. of 2012 Spring Congress on Engineering and Technology (S-CET)*, 2012.

[116] R. L. et. al., "Software engineering for self-adaptive systems: A second research roadmap," *Self-Adaptive Systems*, vol. LNCS 7475, pp. 1–32, 2013.

[117] J. Magee, N. Dulay, S. Eisenbach, and J. Kramer, "Specifying distributed software architectures," in *Proc. of 5th European Software Engineering Conference*, 1995, pp. 137–153.

[118] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41–50, 2003.

[119] J. Kramer and J. Magee, "Self-managed systems: An architectural challenge," in *Proc. of Future of Software Engineering'07*, 2007, pp. 259–268.

[120] M. A. Montes, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, and M. Dorigo, "Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making," *Swarm Intelligence*, vol. 5, pp. 305–327, 2011.

[121] T. Schelling, *Micromotives and Macrobehavior*. New York: Norton, 1978.

[122] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, pp. 1420–1443, 1978.

[123] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of Modern Physics*, vol. 81, pp. 591–646, 2009.

[124] D. Helbing, *Quantitative Sociodynamics. Stochastic Methods and Models of Social Interaction Processes*. Springer, 2010.

[125] C. Sueur, J.-L. Deneubourg, and O. Petit, "From social network (centralized vs. decentralized) to collective decision-making (unshared vs. shared consensus)," *PLoS One*, vol. 7, pp. e32 566:1–10, 2012.

[126] C. A. C. Parker and H. Zhang, "Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem," *IEEE/ASME Transactions on Mechatronics*, vol. 14, pp. 240–251, 2009.

[127] A. Brutschy, A. Scheidler, E. Ferrante, M. Dorigo, and M. Birattari, ""can ants inspire robots?" self-organized decision making in robotic swarms," in *Proc. of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[128] D. Ye, M. Zhang, and D. Sutanto, "Self-adaptation-based dynamic coalition formation in a distributed agent network: A mechanism and a brief survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 1042–1051, 2013.

[129] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings, "Cooperative games with overlapping coalitions," *Journal of Artificial Intelligence Research*, vol. 39, pp. 179–216, 2010.

[130] M. Mares, "Fuzzy coalition structures," *Fuzzy Sets and Systems*, vol. 114, pp. 23–33, 2000.

[131] L. M. Hofmann, N. Chakraborty, and K. Sycara, "The evolution of cooperation in self-interested agent societies: a critical study," in *Proc. of AAMAS'11*, 2011, pp. 685–692.

[132] Y. Wang, A. Nakao, and A. V. Vasilakos, "On modeling of co-evolution of strategies and structure in autonomous overlay networks," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, pp. 17:1–23, 2012.

[133] A. Szolnoki and M. Perc, "Group-size effects on the evolution of cooperation in the spatial public goods game," *Physical Review E*, vol. 84, p. 047102, 2011.

[134] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, pp. 1390–1396, 1981.

[135] E. Fehr and U. Fischbacher, "The nature of human altruism," *Nature*, vol. 425, p. 785C791, 2003.

[136] J. H. Kagel and A. E. Roth, *The handbook of experimental economics*. Princeton University Press, 1995.

[137] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440–442, 1998.

[138] D. Ye and M. Zhang, "A self-adaptive strategy for evolution of cooperation in distributed networks," *IEEE Transactions on Computers*, p. DOI:10.1109/TC.2014.2308188, 2014.

[139] S. Costantini and G. D. Gasperis, "Runtime self-checking via temporal (meta-)axioms for assurance of logical agent systems," in *Proc. of the 7th Workshop on Logical Aspects of Multi-Agent Systems*, 2014.

[140] S. Costantini, "Self-checking logical agents," in *Proc. of LA-NMR'12*, 2012.

[141] ——, "Self-checking logical agents," in *Proc. of AAMAS'13*, 2013, pp. 1329–1330.

[142] J. P. George and et al., "Real-time simulation for flood forecast: An adaptive multi-agent system staff," in *Proc. of Symp. on Adaptive Agents and Multi-Agent Systems*, 2003, pp. 7–11.

[143] J. P. George, B. Edmonds, and P. Glize, "Making self-organizing adaptive multi-agent systems work - towards the engineering of emergent multi-agent systems," in *Methodologies and Software Engineering for Agent Systems*, 2004, pp. 319–338.

[144] M. Mamei, F. Zambonelli, and L. Leonardi, "Co-fields: A physically inspired approach to motion coordination," *IEEE Pervasive Computing*, vol. 3, pp. 52–61, 2004.

[145] M. Camurri, M. Mamei, and F. Zambonelli, "Urban traffic control with co-fields," *Lecture Notes in Artificial Intelligence*, vol. 4389, pp. 239–253, 2007.

[146] H. Karuna and et al., "Emergent forecasting using a stigmergy approach in manufacturing coordination and control," *Lecture Notes in Artificial Intelligence*, vol. 3464, pp. 210–226, 2005.

[147] ——, "A study of system nervousness in multi-agent manufacturing control system," *Lecture Notes in Artificial Intelligence*, vol. 3910, pp. 232–243, 2006.

[148] A. Dury, F. L. Ber, and V. Chevrier, "A reactive approach for solving constraint satisfication problems: Assigning land use to farming territories," *Lecture Notes in Artificial Intelligence*, vol. 1555, pp. 397–412, 1998.

[149] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, pp. 16–27, 2000.

[150] T. H. Labella, M. Dorigo, and J. L., "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, pp. 4–25, 2006.

[151] W. Liu, A. F. T. Winfield, J. Sa, J. Chen, and L. Dou, "Towards energy optimization: Emergent task allocation in a swarm of foraging robots," *Adaptive Behavior*, vol. 15, pp. 289–305, 2007.

[152] Y. Ikemoto, T. Miura, and H. Asama, "Adaptive division-of-labor control algorithm for multi-robot systems," *Journal of Robotics and Mechatronics*, vol. 22, pp. 514–525, 2010.

[153] C. Bernon and et al., "Adelfe: A methodology for adaptive multi-agent systems engineering," *Lecture Notes in Computer Science*, vol. 2577, pp. 156–169, 2002.

[154] ——, "Tools for self-organizing applications engineering," *Lecture Notes in Artificial Intelligence*, vol. 2977, pp. 283–298, 2004.

[155] X. Zhang, V. Lesser, and S. Abdallah, "Efficient management of multi-linked negotiation based on a formalized model," *Autonomous Agents and Multi-Agent Systems*, vol. 10, pp. 165–205, 2005.

[156] X. Zhang, V. Lesser, and T. Wagner, "Integrative negotiation among agents situated in organizations," *IEEE Transactions on Systems Science and Cybernetics: Part C*, vol. 36, pp. 19–30, 2006.

[157] K. H. Low, W. K. Leow, and M. H. Ang, "Task allocation via self-organizing swarm coalitions in distributed mobile sensor network," in *Proc. of AAAI'04*, 2004, pp. 28–33.

[158] W. Li and et al., "Efficient allocation of resources in multiple heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 1775–1788, 2014.

[159] M. G. Kallitsis, G. Michailidis, and M. Devetsikiotis, "A decentralized algorithm for optimal resource allocation in smartgrids with communication network externalities," in *Proc. of IEEE SmartGridComm'11*, 2011.
[160] R. Kota, N. Gibbins, and N. R. Jennings, "Self-organising agent organisations," in *Proc. of AAMAS'09*, 2009, pp. 797–804.
[161] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proc. of WWW'10*, 2010.
[162] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 399: 1–18, 2013.
[163] J. R. Galbraith, "The evolution of enterprise organization designs," *Journal of Organization Design*, vol. 1, pp. 1–13, 2012.
[164] T. Hester, M. Lopes, and P. Stone, "Learning exploration strategies in model-based reinforcement learning," in *Proc. of AAMAS 2013*, 2013, pp. 1069–1076.
[165] M. Mihaylov, Y. L. Borgne, K. Tuyls, and A. Nowe, "Decentralised reinforcement learning for energycefficient scheduling in wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 9, pp. 207–224, 2012.
[166] S. Sengupta, S. Das, M. Nasir, A. V. Vasilakos, and W. Pedrycz, "An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, pp. 1093–1102, 2012.
[167] P. P. Reddy and M. M. Veloso, "Strategy learning for autonomous agents in smart grid markets," in *Proc. of IJCAI'11*, 2011.
[168] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. of INFOCOM'04*, 2004.
[169] C.-H. Yu, J. Werfel, and R. Nagpal, "Collective decision-making in multi-agent systems by implicit leadership," in *Proc. of AAMAS'10*, 2010.
[170] M. Beer, M. dInverno, N. R. Jennings, M. Luck, C. Preist, and M. Schroeder, "Negotiation in multi-agent systems," *Knowledge Engineering Review*, vol. 14, no. 3, pp. 285–289, 1999.
[171] P. Stone, "Learning and multiagent reasoning for autonomous agents," in *Proc. of IJCAI'07*, 2007, pp. 13–30.
[172] M. D'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge, "The dmars architecture: A specification of the distributed multi-agent reasoning system," *Autonomous Agents and Multi-Agent Systems*, vol. 9, pp. 5–53, 2004.
[173] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, 1996, pp. 195–210.
[174] Y. Dimopoulos and P. Moraitis, "Multi-agent coordination and cooperation through classical planning," in *Proc. of IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2006, pp. 398–402.
[175] M. Grzes and P. Poupart, "Incremental policy iteration with guaranteed escape from local optima in pomdp planning," in *Proc. of AAMAS 2015*, 2015, pp. 1249–1257.
[176] S. E. Page, "Self organization and coordination," *Computational Economics*, vol. 18, pp. 25–48, 2001.
[177] J. F. Allard, G. O. Wasteneys, and E. N. Cytrynbaum, "Mechanisms of self-organization of cortical microtubules in plants revealed by computational simulations," *Molecular Biology of the Cell*, vol. 21, pp. 278–286, 2010.
[178] E. A. Gaffney, "The sensitivity of turing self-organization to biological feedback delays: 2d models of fish pigmentation," *Mathematical Medicine and Biology*, 2014.
[179] K.-I. Amemoria and S. Ishii, "Self-organization of delay lines by spike-time-dependent learning," *Neurocomputing*, vol. 61, pp. 291–316, 2004.
[180] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, pp. 230–244, 2008.
[181] Z. Meng, Z. Li, A. V. Vasilakos, and S. Chen, "Delay-induced synchronization of identical linear multiagent systems," *IEEE Transactions on Cybernetics*, vol. 43, pp. 476–489, 2013.

Dayong Ye received his BEng degree from Hefei University of Technology, P.R. China, in 2003 and his MSc and PhD degrees from University of Wollongong, Australia, in 2009 and 2013, respectively. Now, he is a research fellow at Swinburne University of Technology, Australia. His research interests focus on self-organising multi-agent systems, service-oriented systems and cloud computing.

Minjie Zhang received her BSc degree from Fudan University, P.R. China, in 1982 and her PhD in Computer Science degree from University of New England, Australia, in 1996. Now, she is a Professor of Computer Science at the University of Wollongong, Australia. Her research interests include multi-agent systems, agent-based simulation and modeling in complex domains.

Athanasios V. Vasilakos is a professor with the Lulea university of technology in Sweden. He is serving as an editor for many technical journals, such as the IEEE Transactions on Network and Service Management, IEEE Transactions on Cloud Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Cybernetics, IEEE Transactions on NanoBioscience, IEEE Transactions on Information Technology in Biomedicine, ACM Transactions on Autonomous and Adaptive Systems and IEEE Journal on Selected Areas in Communications. He is also general chair of the European Alliances for Innovation.