

UNIVERSIDAD
NACIONAL
DE COLOMBIA

A Computational Justice Model for Resource Distribution in Ad Hoc Networks

Juan Pablo Ospina López

Universidad Nacional de Colombia
Engineering Faculty
Computing Systems and Industrial Engineering Department
Bogotá, Colombia
2019

A Computational Justice Model for Resources Distribution in Ad Hoc Networks

Juan Pablo Ospina López

Dissertation presented as partial requirement to grant the title of :
Doctor of Philosophy - Systems Engineering and Computing

Advisor:
Jorge Eduardo Ortiz, Ph.D.

Research Area:
Applied Computing
TLÖN Research Group

Universidad Nacional de Colombia
Engineering Faculty
Computing Systems and Industrial Engineering Department
Bogotá, Colombia
2019

To Katherine

Acknowledgements

First of all, I want to acknowledge the National University of Colombia. I truly appreciate the excellent education they provided in my undergraduate and graduate programs. Furthermore, I want to recognize the great work of the professors of the Computing Systems and Industrial Engineering Department. Thank you for all your teachings.

I sincerely thank UNAM for the opportunity of doing a research internship at the Complexity Science Center (C3). I learned a lot there and also met very nice people. A Special thanks to Dr. Carlos Gershenson. Your ideas have taken my career to another level. Now I understand I am part of the self-organization of the universe.

Finally, I want to thank all the people who support me during my Ph.D: my family, Katherine, and all my friends. Their infinite love and support are what has brought me this far. I also want to acknowledge all the people who contribute to this research. I have learned a lot from them, and I appreciate all their incredible contributions to my work. Thank you for all.

Abstract

We propose a computational justice model for resource distribution in ad hoc networks using socially inspired computing and agent-based modeling. Ad hoc networks are self-organizing systems in which there is no central controller or other orchestration forms. Therefore, it is not possible to use distribution methods designed for centralized systems that require complete information and where the resource distribution aims to optimize the performance of the whole system without considering the individual goals of the participants. In this work, we used socially inspired computing to formulate a distribution method using stochastic games, institutions, distributive justice, and adaptative computing. We analyzed our proposal through simulation and compared its performance with previous works. The result showed how a distribution method based on computational justice is a potential solution for facing the distribution problem in ad hoc networks. Additionally, we implemented a multi-agent system to evaluate this proposal in a real system and to provide an easy and low-cost platform for developing ad hoc network applications.

Keywords:: computational justice, fairness, socially inspired computing, ad hoc networks, agent-based modelling.

Contents

Acknowledgements	VII
Abstract	IX
List of Figures	XIII
List of tables	1
1. Introduction	2
1.1. Objectives	3
1.2. Our contributions	4
1.3. Chapters organization	5
2. Socially and biologically inspired computing for self-organizing networks	6
2.1. Introduction	6
2.2. Self-organizing communication networks: a historical review	7
2.2.1. Scientific paradigms in communications networks development	8
2.2.2. Complexity signs: self-similar traffic, chaos and scale-free topologies .	8
2.2.3. Control paradigms evolution	9
2.2.4. Current self-organizing communications networks	11
2.3. Networking challenges	12
2.3.1. Scalability	13
2.3.2. Dynamic nature	13
2.3.3. Need for infrastructureless and autonomous operation	14
2.3.4. Heterogeneous architectures	14
2.3.5. Solving problems through collective actions	14
2.3.6. Appropriation and distribution of resources	15
2.3.7. Security and privacy	15
2.4. Biological and social computing inspiring self-organizing networks design .	16
2.4.1. A general methodology	17
2.4.2. Classification and categorizations	19
2.4.3. The need for biological and social self-organizing approaches	22
2.5. Conclusions	23

3. Fairness and computational justice: related work	25
3.1. Introduction	25
3.2. Cooperation patterns and common-pool resources	26
3.3. The distribution problem in self-organizing networks	28
3.4. Fairness and computational Justice: background	32
3.4.1. Classical engineering approach	32
3.4.2. Socially inspired computing approach	35
3.5. TLÖN project: a socially inspired computing system	36
3.6. Conclusions	37
4. A computational justice model for resources distribution in ad hoc networks	38
4.1. Introduction	38
4.2. Self-organizing insitutions for collective actions	39
4.3. Cooperation in stochastic games	41
4.3.1. The prisoner's dilemma	42
4.3.2. The linear public goods game	43
4.4. Distributive justice: Rescher's approach	45
4.5. Adaptive behaviors in self-organizing systems	48
4.5.1. The Borda voting approach	48
4.5.2. Genetic algorithms	49
4.6. Putting all the pieces together	50
4.7. Experimental results and evaluation	51
4.7.1. Experimental method	52
4.7.2. Simulation scenarios	53
4.7.3. Experimental results and performance evaluation	54
4.8. Conclusions	58
5. A self-organizing multi-agent system for ad hoc networks	60
5.1. Introduction	60
5.2. Ad hoc networks: a brief overview	60
5.3. The agent paradigm	62
5.4. A self-organizing multi-agent system for ad hoc networks	63
5.4.1. Artificial agents: functional and structural perspectives	64
5.4.2. Multi-agent system: proposed architecture	68
5.5. Experimental results and performance evaluation	70
5.5.1. Coverage expansion	71
5.5.2. Sensor network	72
5.5.3. Mobile clouds	74
5.6. Conclusions	75

6. Conclusions	76
A. Appendix A: MAS architecture using SADT	78
Bibliography	87

List of Figures

2-1. Control paradigms evolution in artificial systems [18].	10
2-2. A modeling approach for socially and biologically inspired computing	16
2-3. Biologically and socially inspired computing for artificial systems	22
3-1. Cooperation forms in self-organizing open systems [61].	27
3-2. Available resources for mobile clouds [61].	28
3-3. TLÖN: a socially inspired computing system	36
4-1. Stochastic games [144].	42
4-2. An institutional approach for resources distribution	51
4-3. Scenario 1 PD: comparison of allocation methods	55
4-4. Scenario 1 LPG: comparison of allocation methods	56
4-5. Scenario 3: volatile agents for PD and LPG	58
5-1. Ad hoc networks applications	62
5-2. Internal dynamics of a multi-agent system	64
5-3. Agents' architecture	66
5-4. Interaction patterns in multi-agent systems	68
5-5. Second level of analysis with SADT	69
5-6. Coverage expansion: network topology	72
5-7. Coverage expansion: performance evaluation	72
5-8. Sensor networks: topology	73
5-9. Sensor networks: performance evaluation	73
5-10. Mobile clouds: computational justice evaluation	74

List of tables

2-1. Control vs resources: a comparison for artificial systems.	12
2-2. Categorization of biologically inspired models	20
2-3. Categorization of socially inspired models	21
3-1. Application of computational justice in communication systems	33
3-2. A comparison of computational justice model in the context of engineering .	34
3-3. A comparison of socially inspired computational justice model	35
4-1. Ostrom's design principles	40
4-2. Rescher's canons of distributive justice	46
4-3. Scenario 1 PD: comparison of allocation methods	55
4-4. Scenario 1 LPG: comparison of allocation methods	56
4-5. Scenario 2 PD: adaptive strategies	57
4-6. Scenario 2 LPG: adaptive strategies	57
5-1. Hardware features of the nodes	70

1. Introduction

The design and development of future communications networks call for a careful examination of biological and social systems. New technological developments like self-driving cars, wireless sensor networks, drones swarm, Internet of Things, Big Data, and Blockchain are promoting an integration process that will bring together all those technologies in a large-scale heterogeneous network. Most of the challenges related to these new developments cannot be faced using traditional approaches, and require to explore novel paradigms for building computational mechanisms that allow us to face the emergent complexity of these new technological applications.

Furthermore, there is a trend towards the automation of self-organizing networks in which users can be both producers and consumers of resources. For example, in ad hoc networks, energy grids, and sensor networks, there is no centralized controller or other orchestration forms. Instead, the system requires to reach a social agreement to collectivize and allocate the resources of the system. This trend imposes new conditions for facing the distribution problem in self-organizing environments. Since users contribute with their resources, they should be able to use the resources contributed by others. Therefore, the allocation methods need to consider not only the requirements from the system perspective but also the individual goals of each participant. Additionally, this process is too fast, frequent, and complex to be performed by a human operator, requiring a computational approach to face it in the context of self-organizing networks.

Consequently, fairness and computational justice arise as an interdisciplinary research area which is usually related to the allocation of resources. In particular, these models are responsible for distributing the benefits and charges of social cooperation in artificial systems. Although the notion of justice is frequently used in our normal activities, it is difficult to reach a universally accepted definition. This concept is subjective and may be dependent on the context of a social group. Moreover, the analysis of the distribution problem based on the tragedy of the common leads us to the question of how to manage common resources to avoid social dilemmas. Any solution requires an understanding of which processes drive human cooperation and how institutions and norms can be used to promote positive behaviors in a social system.

There are two different approaches of computational justice. First, classical engineering approaches that use objective functions to optimize or evaluate the allocation of resources regarding a set of parameters. However, although there is a diversity of models, they neither provide a well-defined method to achieve a fair distribution process nor are prepared to deal with autonomous entities. The majority of the proposals were developed for centralized systems and require complete information to compute a fairness measure. These approaches are not suitable for facing the distribution problem in self-organizing open systems like ad hoc networks in which distributed control, incomplete information, and uncertain environments are intrinsic aspects of the network operation. Second, socially inspired computing approach which uses social concepts like institutions, distributive justice, and cooperation to create allocation methods based on the behavior of human society. Despite there is no direct application of socially inspiring computing in communication systems, they provide a set of principles that can be used to face the distribution problem in the context of ad hoc networks.

In this work, we argue that it is possible to deal with distribution problems in ad hoc networks using a combination of socially inspired computing and agent-based modeling. This proposal is comprised of: (a) the Ostrom's principles for enduring institutions; (b) a cooperation model based on stochastic games; (c) the Rescher's approach of distributive justice; and (d) adaptation mechanisms through voting systems and genetic algorithms. We take as reference the model introduced in [1]. Essentially, the similarities between the previous works and ours are the Ostrom's approach for institutional analysis and the use of the Rescher theory of distributive justice. However, the cooperation patterns based on stochastic games and the use of genetic algorithms as an adaptation mechanism are totally different. Furthermore, we implemented a multi-agent system using self-organization as a design principle in order to provide an easy and low-cost platform for developing ad hoc networks and also evaluate this proposal in a real system.

1.1. Objectives

To create a computational model based on the concepts of distributive justice that allows us to determine the allocation of resources associated with cooperation processes in ad hoc networks.

- To determine the necessary conditions for distributive justice in ad hoc networks through a systematic literature review.
- To model a knowledge base through a formal language that allows us to represent and reason about the set of established principles.

- To implement the knowledge base in order to determine the distribution of resources in an ad hoc network.
- To verify the operation of the proposed model in at least two test scenarios.

1.2. Our contributions

In this thesis, we proposed a computational justice model for ad hoc networks using the agent paradigm and socially inspiring computing. Our initial motivation for this is based on the fact that in the context of ad hoc networks, there are neither direct application of socially inspiring computing nor suitable computational justice models for facing the distribution problem. This work is organized in four main chapters in which the different stages of the research are presented. In particular, the contributions of each chapter are described below:

- In chapter 2, we have showed that it is possible to use biologically and socially inspired computing for building communications systems. We argue that an abstract analysis of biological and social phenomena can be made to create a conceptual framework for developing a new kind of networking technology. Biologically inspired computing can be used for achieving efficient and scalable networking under uncertain conditions, and socially inspired computing for solving problems through collective actions. Furthermore, we introduced a general method for developing these kind of models and showed how the expected features of the next generation of communications networks turn centralized control into an impractical solution.
- In chapter 3, the distribution problem, cooperation patterns, and some application of common-pool resources were introduced to evaluate their applicability in the context of ad hoc networks. We overview and compare the current models for fairness and computational justice through qualitative analysis. Two different approaches were presented: first, the classical engineering approach, which usually uses a set of measures (quantitative or qualitative) to assess a resource allocation according to a set of parameters. Second, socially inspired computing approach which uses social concepts like institutions, distributive justice, and cooperation to create allocation methods based on the behavior of human society. We present a state-of-the-art of both approaches in the context of communication systems.
- In chapter 4, we proved that it is possible to deal with the distribution problem in ad hoc networks using a combination of socially inspired computing and agent-based modeling. In particular, this proposal is composed of: the Ostrom's approach for institutional analysis, cooperation patterns through stochastic games, the Rescher's idea of distributive justice, and adaptive mechanisms through voting systems and genetic algorithms. The result showed how an allocation process based on computational justice is

an alternative to face the distribution problem in environments formed by autonomous agents. It is important to mention that this work was developed on the foundation of self-organizing electronic institutions established by Pitt et al. [1], in which they showed the potential value of the distributive justice and social institutions as part of the future technological systems. Also, we used a variation of the equality canon of justice that allows to improve the response of our proposal to selfish behaviors in the system.

- In chapter 5, we overview some of the most significant applications of the ad hoc networks and their connection with the agent paradigm. Two different approaches for creating artificial agents were presented: first, the functional approach that allows us to model and implement different types of behavior based on the observed actions of the agents. Second, the Sartrean model, which provides us a structural perspective based on biologically and socially inspired computing. Additionally, we implement a multi-agent system as a computational framework for developing ad hoc network applications based on the agent paradigm. We aim to provide an easy and low-cost platform for implementing ad hoc networks using self-organization as a design principle.

Some contents of this thesis have already been presented in [2, 3, 4, 5, 6, 7, 8, 9] and also there are some papers in evaluation process. Each chapter was written with the intention of becoming a research article.

1.3. Chapters organization

This thesis is organized as follows; this introduction describes the research problem and shows its significance in the context of ad hoc networks. Also, the objectives and the thesis contributions are presented. Chapter 2 shows an overview of the challenges of communication systems and explains how control paradigms have evolved in the last decades. Additionally, we introduce a general method for developing computational models based on biologically and socially inspired computing. Chapter 3 describes the distribution problem and presents related work regarding fairness and computational justice in the context of communication systems. Chapter 4 presents our proposal. A computational justice model based on the Ostrom institutional approach, stochastic games, and adaptative techniques is introduced and evaluated through simulation analysis. Chapter 5, describe a multi-agent system implementation as a framework for developing applications for ad hoc network through the agent paradigm. Chapter 6 presents both the conclusions and some guidelines for future research.

2. Socially and biologically inspired computing for self-organizing networks

2.1. Introduction

During the last decades, the number of services and technologies available for networking applications has increased significantly. These developments have shown a direct relationship with different aspects of human society like the economy, education, politics, and quality of life. Computational devices seem to be ubiquitous and are present in almost all aspects of our daily activities. This trend is promoting a technological integration that has already gone beyond what traditional networking paradigms can do regarding scalability, dynamic environments, heterogeneity, and collaborative operation. Consequently, these conditions impose several challenges for building the future networking technologies and show the need to explore new engineering approaches.

The future communication systems will be composed of ubiquitous and self-operating devices that will transform our immediate environment into an intelligent computational system. New technological developments like self-driving cars, wireless sensor networks, drones swarm, Internet of Things, Big Data, and Blockchain are promoting an integration process that will bring together all these technologies in a large-scale heterogeneous network. All these applications involve a set of autonomous components (with possibly conflicting goals) interacting asynchronously, in parallel, and peer-to-peer without a centralized controller. They should be easily accessible by users and operate with minimum human intervention.

Accordingly, all computational devices should be able to operate autonomously and collaborate with others to offer services through collective actions. The future communication networks will require high levels of self-organization for both, face challenges related to scalability, heterogeneity, and for minimizing centralized control and human intervention during the processes of planning, deployment, and optimization of the network. Indeed, these requirements cannot be faced using traditional approaches because they are not able to manage

scale, heterogeneity, and complexity of the future networking applications, and as a result, it is necessary to explore novel paradigms for designing and implementing new models that can operate under those conditions.

In this chapter, we aim to introduce and overview the biologically and socially inspired computing used as technological solutions in networking and artificial systems. The principal idea is to show that it is possible to create analogies between living and artificial systems that enable us to inspire mimetic solutions (biological, social, economic or political) and translate those principles into engineering artifacts. Living systems show desirable properties like adaptation, robustness, self-organization, and learning, all of them required to handle the complexity of the future networking systems. In this regard, we can analyze biological and social phenomena as a source of inspiration for new technological developments; biologically inspired computing for achieving efficient and scalable networking under uncertain environments, and socially inspired computing for increasing the capacity of a system for solving problems through collective actions. In this work, we expect to provide a better comprehension of the opportunities offered by these models and encourage other researchers to explore these approaches as part of their future work.

The rest of this chapter is organized as follows: in Section 2.2, we present a historical review of the scientific and technological development of communication systems. In Section 2.3, we summarize the most challenging issues of the next generation of communication networks from the perspective of biologically and socially inspired computing. Section 2.4 introduces a general method for developing these models; the main idea is to expose how to create a technological solution from properties and behaviors observed in living systems. Section 3.6 concludes the chapter.

2.2. Self-organizing communication networks: a historical review

In this Section, the need for using self-organization as a control paradigm for the next generation of communication networks is discussed. First, we provided a historical review of the scientific paradigms used for studying and building communications systems. Second, we show complexity signs related to traffic, topologies, and chaotic behaviors because of interactions among users, nodes, and applications. Third, a comparison of the current control and management paradigms used for designing, controlling, and developing artificial systems is presented. Finally, we depict some properties required for the future communications systems based on self-organizing properties.

2.2.1. Scientific paradigms in communications networks development

Traditionally, the scientific paradigm used for communications networks development has been reductionism. Engineers conceived communication systems as a hierarchical structure that allows offer services through protocols and distributed algorithms; each layer was studied individually, and a communication interface among them was used to provide functionalities during the network operation [10]. Devices, protocols, and applications were designed separately, and linear behavior in the whole system was expected. This idea arose from the first mathematical models used for planning and dimensioning communications systems, in which engineers used stochastic models and queue theory to compute the average traffic and assign resources according to the users' demands [11, 12]. This approach played an essential role in traditional telephone networks in which there was only one service, and the performance required for all users was the same. Thus, it was easy to combine traffic flows and take advantage of their homogeneous features for analytical purposes. However, an increasing amount of networking technologies and also more complex software applications changed the linear behavior expected inside communications networks [13].

During the last decades, an integration of services and technologies available for networking applications has occurred. Nowadays, it is possible to find data transfer, online games, video, email, e-commerce, and browsing, working on the same network infrastructure [14]. Also, we can find different transmission technologies like wired connections, optical fiber, IEEE 802.11, WiMAX or Bluetooth, and the performance required for each application (bandwidth, delay, and error handling) is different every case [15]. As a consequence, this increasing number of services and technologies changed the design principle on which engineers based the networking development: linearity. Communication networks do not have linear behaviors anymore, and it is needed to analyze them as complex systems if we want to design algorithms and control mechanisms capable of operating in a dynamic environment with non-linear properties [16, 17].

2.2.2. Complexity signs: self-similar traffic, chaos and scale-free topologies

Because of technologies and services integration, communications networks started showing complexity signs like self-similar traffic, chaotic behaviors, and scale-free topologies. Although none of these properties were in the initial conceptual framework used by engineers for design and building communications systems, nowadays, there is enough evidence to consider them as an inherent part of the communication networks. A brief overview of these complexity signs is exposed below.

Usually, traffic is modeled as a stochastic process that shows the amount of data moving across a network and establishes a measure to represent the demand that users imposed on the network resources. Both requests per time unit and the incoming packets have been modeled as sequences of independent random variables (call duration, packet lengths, file sizes, etc.) to make easier their analytical treatment [12]. However, the correlation among these variables persists through several time scales and has a significant impact on the network performance [18, 19]. It is important to mention that self-similarity is not a property of traffic sources; it arises as emergent behavior from interactions among users, applications, and networking protocols. Besides, traditional traffic models based on Poisson processes has proven not to be suitable to describe traffic patterns in modern communications networks [11, 20].

Similarly, chaotic behaviors take place in dynamical systems that are high-sensitivity to initial conditions; small differences in the system states can produce a significant number of different outcomes. Chaos theory studies these behaviors and tries to deal with the apparent randomness present in strange attractors, feedback loops, and self-similarity. For example, in communications networks, this behavior appears through interactions between TCP protocol and the RED algorithm used for queue management [21]. Other examples are presented in [22] in which chaos appears in the profile of daily peak hour call arrival and daily call drop of a sub-urban local mobile switching center or in which chaotic patterns serve as a mobility model for an ad hoc network [23]. More examples can be found in [24, 25, 26].

Finally, the scale-free property is another complexity sign that suggests self-similarity patterns in terms of the network topology [27]. The structure of the network has nodes with more connections than others, and it follows a power-law distribution. This pattern was found in the late 1990s when a part of the *World Wide Web* was mapped in a moment of internet connection [28]. This phenomenon could be explained by analyzing the evolution of communications networks in terms of their physical and logical topologies according to the preferential connectivity principle [29]. If a web page is created, it is reasonable to assume that links to highly connected sites like Google, social networks, services companies, etc., will be added. Also, the physical topology of the internet is defined by the economic and technological requirements of Internet Service Providers (ISP) [20].

2.2.3. Control paradigms evolution

All artificial systems, including communications networks, use management, and control processes to regulate their behaviors. The management process consists of manipulating subsystems, parameter updates, and verify the system state. On the other hand, control is about

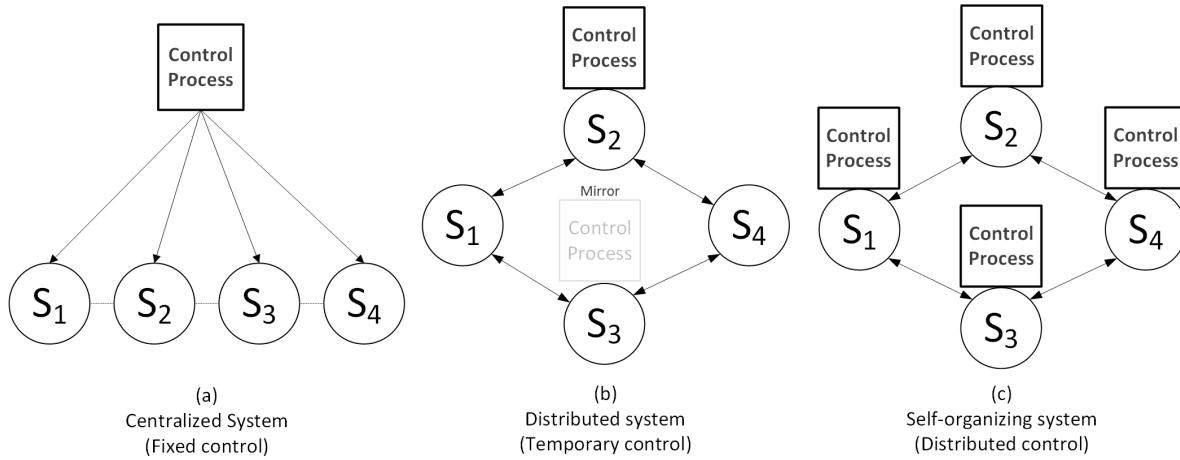


Figure 2-1.: Control paradigms evolution in artificial systems [18].

feedback and run-time control according to variations in the environment. Both processes define the routines to maintain, operate, and adapt the system during operation time. Figure 2-1 shows a historical review of the current control paradigms for artificial systems [13, 18].

Initially, communications networks were composed of a single device and some remote terminals. There was a single control process, and all parameters required for the network operation, e.g., addresses, access privileges, and resources were pre-configured by default. Changes in topology and applications were possible but required a complete manual configuration of the system [18]. Figure 2-1a presents an example of these control paradigm through a hierarchical architecture; the root shows the control process and the leaves the subsystems it can handle. For instance, traditional telephone networks and client/server applications are classic examples of this approach [11]. Even though there are others control schemes, centralized systems are still the preferred solution due to its simplicity and effectiveness; if only a few well-known subsystems have to be managed, there is no need for the high computational cost of distributed algorithms or possibly less deterministic self-organizing methods [30, 31].

The next paradigm is the distributed control [32, 13]. Distributed systems are composed of a set of independent nodes that works as a single coherent system. In this case, a logical abstraction is deployed as middleware in each device to hide the internal structure and the communication process for the application layer. Figure 2-1b shows a scheme for this paradigm. Although the control process works in a centralized way, it is possible to locate it dynamically inside any node to improve the fault tolerance and achieve better use of the resources. Cellular networks, distributed databases [33], orchestration software [34], and multi-agent systems like JADE [35] are examples of this control paradigm. Distributed systems offer several advantages to operate and combine resources from different nodes. Ho-

ever, some issues like impossible synchronization and overhead for resource management show the limits of this approach [36, 13]. The need to maintain complete information about the system state and handle changes related to configuration and topology is an expensive computational task in highly dynamic environments [13].

Finally, we have self-organizing systems [31, 16]. In this approach, the management and control process is completely distributed, i.e., each sub-system has its own control process. The functionalities and the system structure arise as emergent behaviors from interactions among elements. Similarly, the goals of the system should not be designed, programmed, or controlled by default; the components should interact with each other until they reach the expected configuration. Self-organizing control is flexible, adaptive, robust, and scalable; it does not need perfect coordination and can operate in dynamic environments [37]. Since each component is autonomous, it is necessary to develop additional mechanisms to promote cooperation, coordination, and synchronization among the system components. It is important to mention that self-organization is not a human invention; it is a natural principle that has been used for designing, building, and controlling artificial systems, and face limitations of centralized and distributed approaches [13]. Examples of this control paradigm can be found in Smart Grids [38], communication networks [39], transportation systems [40], and logistical processes [41].

Although self-organization increases scalability, it also causes less deterministic behaviors. The system predictability is reduced due to self-organized control. Nevertheless, this is not a real disadvantage in a dynamic system with non-linear properties in which an approximate solution can be useful. Additionally, we are in a transition process from distributed to self-organizing systems due to changes in the network architectures, new computational technologies, and the need to build large-scale communication systems [15, 14]. To sum up, Table 2-1 describes the relationship between resources and control according to the different paradigms presented above.

2.2.4. Current self-organizing communications networks

The increasing use of mobile devices, pervasive computing, wireless sensor networks (WSNs), and cloud computing establish new requirements for future communications systems (See Section 2.3). New applications like self-driving cars [42], drones swarm [43], Internet of Things [44], Big Data and Blockchain [45] are promoting a technological integration that will bring together all these applications in a large scale heterogeneous network. As a result, the future networking applications will require high levels of self-organization for both, face challenges related to scalability, heterogeneity, and dynamic environments, and minimize centralized control and human intervention during the processes of planning, deployment,

Table 2-1.: Control vs resources: a comparison for artificial systems.

Control paradigms for artificial systems		
Centralized Systems	Resources Control	Centralized Centralized
Distributed systems	Resources Control	Distributed Centralized
Self-Organizing systems	Resources Control	Distributed Distributed

and optimization of the network. These challenges may be faced through a set of networking functionalities based on self-organizing properties [46, 14, 15]:

- *Self-configuration*: in this context, configuration refers to how the network is set up. Nodes and applications should configure and reconfigure themselves automatically under any predictable or unpredictable condition with minimum human intervention. Self-configuration expects to reduce the effects of networking dynamics to users.
- *Self-deployment*: preparation, installation, authentication, and verification of every new network node. It includes all procedures to bring a new node or applications into operation. Also, self-deployment try to find strategies to improve both coverage and resource management in networking tasks.
- *Self-optimization*: it refers to the use of measurements and performance indicators to optimize the local parameters according to global objectives. It is a process in which the network settings are autonomously and continuously adapted to the network environment regarding topology, resources, and users.
- *Self-healing*: execution of routines that keep the network in the steady-state and prevent problems from arising. These methods can change the configuration and operational parameters of the overall system to compensate for failures.

2.3. Networking challenges

Indeed, the majority of the requirements for the next generation of communication networks cannot be faced using traditional approaches [47, 14, 31]. In this Section, we present some of those challenges and their possible relationship with biological and social phenomena. It is important to mention that this Section is not a full reference of challenges in networking but could be seen as a list we can address through biologically and socially inspired computing.

2.3.1. Scalability

One of the most desirable properties in communication networks is the capacity to increase the network size and be able to receive new nodes and applications without affecting the quality of the services [5]. This property, known as scalability, is one of the leading challenges in protocols design, and it is a requirement for building large-scale communication systems. Scalability can be measured regarding applications, users, physical resources, and the network's ability to react correctly to unexpected conditions [36]. For example, wireless sensor networks usually need to collect data from several hundred sensors, and during this process the capacity of the network can be easily exceeded, causing loss of packets, low network reliability, and routing problems [48].

Furthermore, the decision process required to operate a large-scale network is too fast, too frequent, and too complex for being handled by human operators. As a result, network components need to self-organize by themselves across different scales of time and space to adapt their behavior to any variation in the network size [31, 49]. Fortunately, there are many biological and social systems with self-organization mechanisms we can learn from to inspire the design of scalable systems [50]. For instance, data dissemination based on epidemic spreading [47], routing protocols based on Ant Colony Optimization (ACO) [51], and trust and reputation models for controlling free-riders may help to face challenges related to large-scale networking [52].

2.3.2. Dynamic nature

Unlike traditional communication networks in which infrastructure and applications were static, the future networking schemes will be highly dynamic regarding devices, users, resources, and operating conditions [39, 14]. For example, the network topology may change according to different mobility patterns, and applications will need different levels of performance concerning bandwidth, delay, and error handling [53]. Also, cognitive radio allows us to configure the spectrum dynamically through overlapping spectrum bands, and users may decide what will be their role in the network due to the absence of centralized control [54]. Additionally, the increasing autonomy in the network components may cause unexpected behaviors, turning into a difficult task to predict the temporal evolution of the system. Under these conditions, self-organizing protocols are essential to improve adaptation, robustness, and face challenges related to highly dynamic environments [18, 31].

2.3.3. Need for infrastructureless and autonomous operation

The current levels of heterogeneity in communication systems in terms of users, devices, and services become centralized control, an impractical solution [18, 14]. Moreover, there is another trend towards automation in which networking applications require to operate with minimum human intervention. For example, drone swarm [43], delay-tolerant networks [55], sensor networks [48], and cognitive radio [39], demand networking protocols that can operate without centralized control, recover from failures, and deal with highly dynamic environments. In order to address these needs, networking protocols could be equipped with self-organizing mechanisms observed in biological and social systems to develop autonomous applications and decrease the level of centralized control required for the network operation [47, 56].

2.3.4. Heterogeneous architectures

Future communications networks require integrating several technologies through internet-based platforms. Given the diverse range of networking components and the numerous interactions among them, it is reasonable to expect complex global behaviors. The next generation of networking applications will be composed of WSNs, ad hoc networks, wireless fidelity networks, VANETs, etc., all of them working on a large-scale communication system. [15, 14]. For instance, one of the emerging and challenging future networking architectures is the Internet of things (IoT) [57]. This paradigm includes the pervasive presence of network devices that through wireless connections, can communicate among them and transform our immediate environment into an intelligent large-scale computational system. Also, Wireless Mesh Networks and WiMAX are expected to be composed of heterogeneous devices and protocols [39].

Heterogeneity needs to be understood, modeled, and managed regarding technologies, users, and applications if we want to take advantage of large-scale heterogeneous networks [47]. Therefore, we can analyze living systems with high levels of heterogeneity and use them to inspire technological solutions. For example, biological and social phenomena show stable behaviors through the cooperation of a heterogeneous set of subsystems, e.g., nervous system, immune system, and normative social systems. This functionality is called homeostasis and can be used for designing computational mechanisms to face challenges related to heterogeneity [58].

2.3.5. Solving problems through collective actions

A standard requirement in self-organizing communication networks is to produce coordination, cooperation, and synchronization among the network components to achieve individual

and collective goals. This process can be understood as a requirement to solve problems through collective actions, in which the accomplishment of tasks depends on interaction and interoperation of unreliable and conflicting components [59]. Likewise, due to the absence of centralized control, the network is instead relying on self-organization mechanisms to produce the system functionalities. These models are useful for resource provisioning in grid computing [60], cooperation in mobile clouds [61], platooning in vehicular networks [62] and coordination in drone swarms [43].

Collective actions are necessary to construct new levels of social organization; multicellular organisms, social insects, and human society use it to take advantage of skills and knowledge of others to achieve collective benefits [63]. Although this is a common phenomenon in living systems, it is important to mention that human society has more complex collective action patterns than other species, and we can use them as a source of inspiration for engineering developments. For example, computational justice models could be used for appropriation and distribution of resources in mobile clouds, and ad hoc networks [60], cooperation models for controlling free-riders, and promote collaborative work among network components [52]. Also, collective behaviors from biological systems like firefly synchronization and swarm intelligence could improve routing and network optimization [31, 47].

2.3.6. Appropriation and distribution of resources

One advantage offered by the next generation of communication networks is the opportunity to share resources among nodes, users, and services through the combination of wireless technologies, mobile devices, and the network capacity to operate as a self-organizing system. For example, a mobile cloud allows to exploit distributed resources inside a network if they are wirelessly connected; energy, storage, communication interfaces, and software applications can be exchanged, moved, augmented and combined in novel ways [61]. Also, grid and cloud computing provided an infrastructure based on common-pool resources to support on-demand computing applications [60]. As a consequence, optimal mechanisms for resource appropriation and distribution are required [64, 65]. This process may be in a stochastic or deterministic manner, and the network components need to self-organize themselves to achieve a distributed resources operation. In this regard, several challenges related to how to carry out a sustainable cooperation process in environments composed of potentially selfish components arise. One solution could use electronic institutions and social capital as a way to increase the capacity of the network to use collective actions. Applications of this approach can be found in Smart Grids [66], VANET's [67], and multi-agent systems [68].

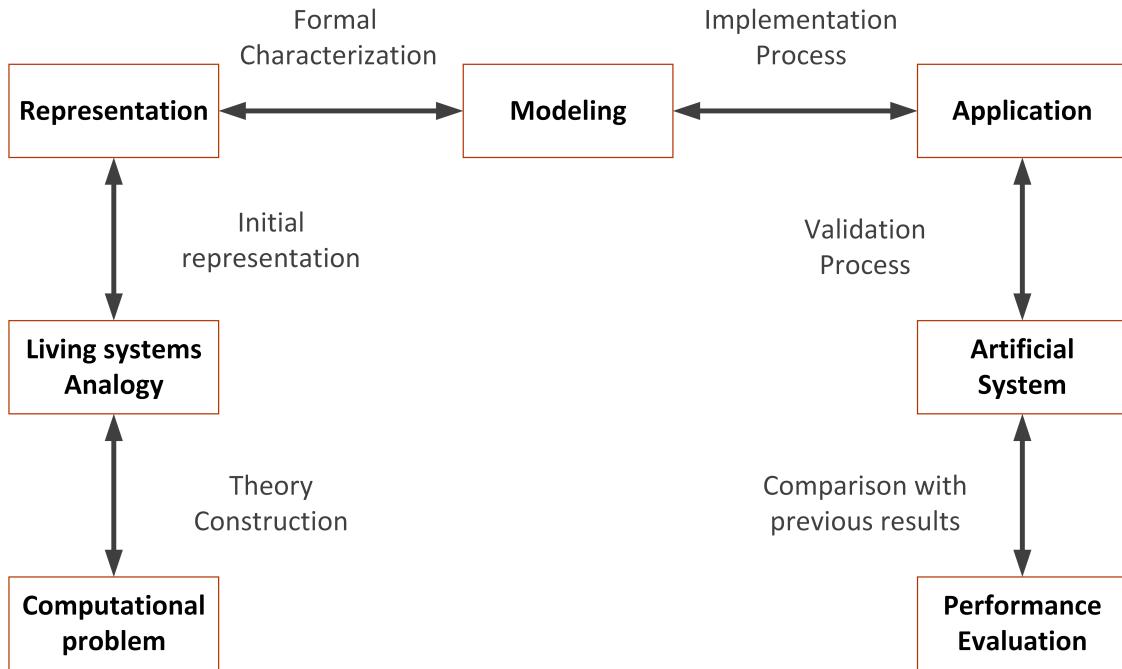


Figure 2-2.: A modeling approach for socially and biologically inspired computing

2.3.7. Security and privacy

Since the networks become flexible, attackers can get sensitive information analyzing the messages embedded in communications channels and relay nodes [15]. Also, according to mobility patterns, the network topology may change in dynamical and unpredictable ways changing routing tables and increasing the risks of exposing crucial private information [14]. As a result, there are several security challenges, such as a denial of service, black hole, resource consumption, location disclosure, wormhole, and interference [39]. For instance, the future internet of things will transfer a significant amount of private information through wireless channels, and security protocols need to defend malicious attacks to provide a relatively secure network environment [69]. One solution could use game theory to address situations where multiple players with contradictory goals or incentives compete among them. Many biological and social systems have inspired solutions to deal with security and privacy issues. For example, artificial immune systems for anomaly and misbehavior detection and trust and reputations models to control free-riders and selfish behaviors [47].

2.4. Biological and social computing inspiring self-organizing networks design

One purpose of this work is to introduce and overview the biologically and socially inspired models used as technological solutions in networking and artificial systems. The main idea is to show how an abstract analysis of living systems (biological, social, economic, or political) can be made to develop computational models that may provide a suitable conceptual framework for technological developments. According to this purpose, this Section is organized as follows: first, we present a general method for developing computational models inspired by biological and social phenomena. Second, we try to classify them and present some selected examples to motivate their applications in the current networking developments. Finally, we depict the need for both biologically and socially inspired computing in the next generation of communications systems.

2.4.1. A general methodology

The modeling approach presented below should not be seen as a general principle, but it may work as a guideline to design algorithms and protocols for artificial systems. It is essential to mention that the proposed steps are not new and have been used by many researchers during the last years [70, 71, 72, 47, 73]. However, we try to take the essential parts of the approaches presented by Dressler for biologically inspired networking [47], Pitt for socially inspired computing [56], and Gershenson for designing and controlling of self-organizing systems [31]. Our aim is to show the necessary steps for developing biologically and socially inspired models, and also present how they may have a remarkable impact on technological developments. Figure 2-2 presents the steps included in this methodology. It starts with the required system functionality, i.e., what the system should do, and enables the designer to produce a protocol or an algorithm that fulfills those requirements. Also, it is not necessary to follow these steps in order; according to the designer's needs, it is possible to return to an early step to make any necessary adjustments.

Identification of analogies between living and artificial systems

In the first step, an analogy between living and technological systems must be made to identify similar patterns that help to understand and propose new computational solutions [47, 56]. Analogies are the tools of the comprehension; people understand new concepts by relating them to what they already knew [74]. If we chose the right analogy, the model reaches a level of abstraction that allows people foreign to the problem, get a better understanding through a well-known vocabulary. Also, create analogies among different systems will enable us to inspire mimetic solutions (biological, social, economic, or political) and translate those

principles into engineering artifacts. However, analogies could have a limitation regarding expressiveness; using a specific description to represent a problem, may limit its comprehension if the analogy is not good enough. Therefore, you can not use every analogy you know; it is necessary to master the selection process to get access to new interpretation tools.

Representation

In this step, a pre-formal representation that relates the observed biological or social phenomenon with a technological problem is developed. The designer should always remember the distinction between the model and the modeled; there are many representations of a system, and it is not possible to say one is better than another independently of a context [31, 56]. Similarly, the initial representation can be made in natural language or through any tool that allows us to describe variables, abstraction levels, granularity, and interactions among components.

Although there is a wide diversity of systems, we can use a general method for developing an initial representation [31]. First, we need to divide the systems into components and identify their internal goals. Second, since the number of components may increase the complexity of the model, we should group them according to their dynamic, and analyze the most important based on the problem requirements. Finally, the designer should consider at least two abstraction levels to capture emergent properties and possible collectives behaviors. Nevertheless, if the initial description has just a few elements, probably the system is predictable, and we could get a better understanding through traditional approaches [16].

Modelling

In science and engineering, models should be as simple as possible and predict as much as possible; they should provide a better understanding of problems and not complicate them unnecessary [31, 75, 76]. Also, the quality of the model is related to the analogies we chose to describe the system; if the model becomes impractical, the selected representation should be carefully revised [74]. Implementation issues should not drive this stage because its primary goal is to achieve a clear understanding of the problem through a formal analysis of biological and social phenomena.

Furthermore, this stage should specify a control paradigm that ensures the expected behavior of the system. Since we are interested in self-organizing properties, the control mechanisms need to be internal and distributed. Given these conditions, several approaches like action languages, modal logic, game theory, and agent-based modeling have been extensively used to model complex systems and may help during this process. Finally, the expected result of

this stage is a formal characterization that will enable us to translate biological and social principles into computational protocols [56].

Application

This step aims both to translate the current model into computational routines, and tune its parameters through different test scenarios. This process should be made from general to particular. Usually, little details take time to develop, and sometimes we will require an ideal scenario to test the central concepts involved in the model (for example, through simulation techniques) [31, 75]. Particular details can influence the system behavior, and they should not be included meanwhile their mechanisms and effects are not understood. According to the application results, modeling and representation stages should be improved.

Moreover, to get algorithms or protocols with acceptable computational tractability, probably we need some degree of simplification in the concepts involved in the model. However, it is a good practice to get as transparent as possible an idea of what is going to be simplified; any simplification that needs to be done should be carried out carefully with the purpose of not to dismiss essential parts of the model [56]. In an ideal scenario, the application stage should not be constrained by considerations of computational tractability.

Performance evaluation

The purpose of this step is to measure and compare the performance of resulting algorithms or protocols with the performance of previous results. This is an essential part of the method because it allows us to integrate our results with the current scientifical and engineering developments. Also, if the system has multiple designers, they should agree on the expected functionality of the system [31, 47]. According to the performance evaluation, the efforts to improve the model should continue as long as possible and even return to an early step to make any necessary adjustments.

2.4.2. Classification and categorizations

Most of the proposed solutions for self-organizing networks are based on biologically inspired computing, and they have successfully solved problems related to routing, synchronization, security, and coordination [47]. However, there is a new kind of socially inspired computing coming up; human society has many self-organizing mechanisms that we can learn from to enhance the capacity of artificial systems to solve problems through collective actions [56, 49, 59]. Not only these models are useful to face the tension between the individual and collective rationality, but also they help to answer questions like: are the cooperation processes sustainable? Is the resource distribution efficient and fair? Can a set of rules evolve

Table 2-2.: Categorization of biologically inspired models

Biological principle	Application fields in networking	POE	Selected references
Swarm intelligence and social insects	Distributed search and optimization; routing in computer networks especially in MANETs, WSNs, and overlay networks; task and resource allocation	E	[77], [78], [79], [80], [81], [82], [83]
Firefly synchronization	Robust and fully distributed clock synchronization	E	[84], [85], [86], [87], [88], [89], [90]
Artificial immune system	Network security; anomaly and misbehavior detection	E	[91], [92], [93], [94], [95], [96], [97]
Epidemic spreading	Content distribution in computer networks (e.g. in DTNs); overlay networks; analysis of worm and virus spreading	PE	[98], [99], [100], [101], [102], [103], [104]
Evolutionary computing	Optimization, cooperation strategies, adaptation to dynamic conditions.	PO	[105], [106], [107], [108], [52], [109], [110]

autonomously in an artificial system? Socially inspired computing tries to answer these questions through a formal analysis of social phenomena. It is important to mention that neither all socially and biologically inspired models are related to self-organizing properties, nor all self-organizing behaviors arise from living systems. However, this work focus on computing models with distributed and internal control related to social and biological systems. An overview of these models is presented in the following subsections.

Biologically inspired computing

Biological systems exhibit a wide range of desirable characteristics, such as evolution, adaptation, fault tolerance, and self-organizing behaviors. These properties are difficult to produce using traditional approaches, and make it necessary to consider new methods [111]. Thus, the purpose of biologically inspired computing is design algorithms and protocols based on biological behaviors that allow artificial systems to face challenges related to optimization, collective behavior, pattern recognition, and uncertain environments [112, 113]. Typical examples of these models can be found in swarm intelligence, firefly synchronization, and evolutionary algorithms [47, 114]. Table 2-2 shows a summary of biologically inspired models successfully used in networking.

Furthermore, if we analyze living organisms, three different organization levels are found: Phylogeny (P), Ontogeny (O), and Epigenesis (E) [115]. First, Phylogeny is related to the temporal evolution of the genetic program. This process is fundamentally non-deterministic and gives rise to the emergence of new organisms through recombination and mutation of the genetic code. Second, Ontogeny is related to the development of a single individual from its genetic material. Finally, Epigenesis is concerned about the learning process in which an

Table 2-3.: Categorization of socially inspired models

Social principle	Application fields in networking and artificial systems	Selected references
Trust	Cooperation mechanisms in self-organizing artificial systems: Ad hoc networks, WSNs, Smart Grids, Multi-Agent Systems	[116], [117], [118], [119], [120], [121]
Justice	Resources distribution and allocation; Smart Grids, VANETs, multi-agent systems; social dilemmas	[65], [60], [59], [122], [38], [64], [67]
Norms and institutions	Evolution of norms. Institutions as a mechanism for collective actions. Self-organizing open systems	[123], [124], [125], [126], [127], [128]
Negotiation	Resources negotiation in ad hoc network and multi-agent systems.	[129], [130], [131], [132]

organism can integrate information from the outside world through interactions with the environment. The distinction among these categories cannot be easily drawn and may be subject to discussion.

POE model can be used in the context of engineering to classify biologically inspired models and identify possible directions for future research [113]. We can understand the POE model as follows: Phylogeny involves evolution; Ontogeny involves development, and Epigenesis involves learning. In this regard, evolutionary computing can be seen as a simplified artificial counterpart of Phylogeny in nature. Multicellular automata, self-replicating, and self-healing software are based on ontogeny properties. For example, when a program can produce a copy of its code or regenerate parts of itself to compensate failures. Finally, artificial neural networks and artificial immune systems can be seen as examples of epigenetic processes. In Table 2-2, a classification of the biologically inspired models according to POE model is presented.

Socially inspired computing

Pitt, Jones, and Artikis introduced social inspired computing as a way to create mechanisms that allow artificial systems to solve problems through collective action [56]. Even though this is not the first attempt to use social models in computer science [72, 73], from the author's knowledge is the first proposal that presents a systematic method to develop them. These models are useful in systems formed by a set of co-dependent components in which there is a tension between the individual and collective rationality [133, 59]. In such systems, the achievement of individual and collective goals depends on possible unreliable and conflicting components, interacting in the absence of centralized control or other orchestration forms.

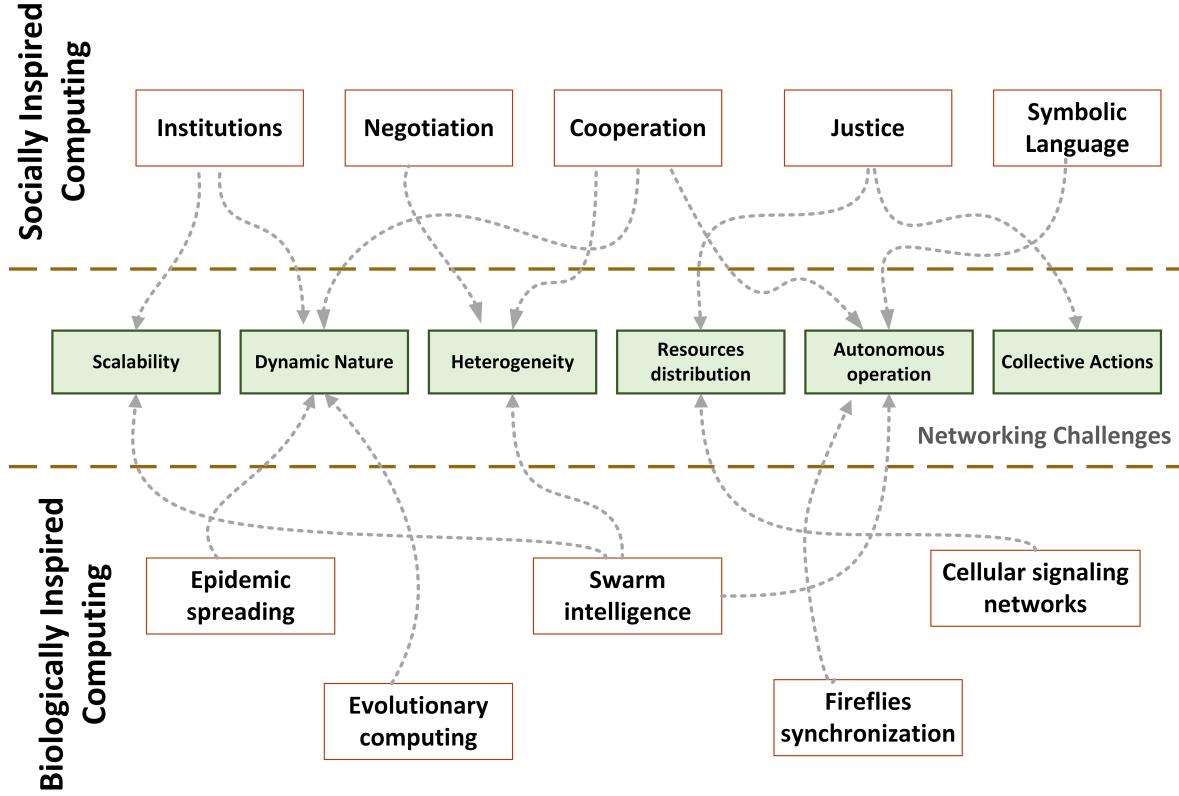


Figure 2-3.: Biologically and socially inspired computing for artificial systems

Although biological processes are the foundation of social systems, they are not the core of sociability. Although both living organisms and societies can be considered as meta-systems, the difference between them is the level of autonomy in their components; while the units of an organism have little or no independence, those of social systems have a maximum level of autonomy. As a result, new kinds of self-organizing phenomena appear, and it is valuable to make a difference between biologically and socially inspired computing. On the other hand, human society has more complex social patterns than other species; cooperation, institutions, symbolic language, and justice could be useful to inspire computational mechanisms that allow us to translate these principles into technological artifacts [49, 72]. In Table 2-3, a summary of socially inspired models successfully used in ad hoc networks, smart grids, and multi-agent systems is presented.

2.4.3. The need for biological and social self-organizing approaches

The design and development of communication networks, as well as all self-organizing artificial systems, call for a careful examination of biological and social concepts. In this Section, we present the relationship between the networking challenges presented in Section 2.3 and

the biologically and socially inspired models that we may use to deal with them. Although both biological and social inspired models exhibit self-organizing patterns, in each case, their goals are different. Biologically inspired computing try to achieve efficient and scalable networking under uncertain environments, and socially inspired computing is useful for solving problems through collective behaviors. Therefore, the combination of these two approaches allows us to develop communication networks not only enough robust and adaptive to be able to operate in highly dynamic environments, but also with the capacity to use collective actions for solving complex problems. In Figure 2-3 the relationship between the biologically and socially inspired models and the networking challenges presented in Section 2.3 is shown.

In general terms, a self-organizing network is a dynamic system of many agents (which may represent nodes, services, applications, users) working in parallel, always acting and reacting to what the other agents are doing. The control process is highly dispersed and decentralized, and any expected behavior in the network needs to arise from competition, cooperation, or coordination among network components [134]. Biological and social systems have dealt with similar situations for thousands of years, and we can learn from them to develop new types of computational solutions. Although biologically inspired computing has been successfully used during the last years, at this moment, it is necessary to design technological artifacts able to solve problems through collective actions. Therefore, socially inspired computing turns into an opportunity for the next generation of artificial systems, giving us a route to include these properties in future engineering developments.

2.5. Conclusions

In this chapter, we have shown that it is possible to use biologically and socially inspired computing for building communications systems. We argue that an abstract analysis of biological and social phenomena can be made to create a conceptual framework for developing a new kind of networking technology. Biologically inspired computing can be used for achieving efficient and scalable networking under uncertain conditions and socially inspired computing for solving problems through collective actions. The combination of these two approaches enables us to develop communication networks not only enough robust and adaptive to operate in highly dynamic environments but also with the capacity to use collective behaviors for solving complex problems.

Furthermore, we showed the challenges of the next generation of communication networks from the perspective of biologically and socially inspired computing; we introduced a general method for developing these models. We presented an overview in tables 2-2 and 2-3. Also, we argue that the expected features of the next generation of communications networks be-

come centralized control an impractical solution, and as a result, self-organization will take an essential role in the future networking developments.

Despite the considerable amount of ongoing advances in biologically and socially inspired computing, the research community is still quite young. There are many challenges that we need to face if we want to integrate these models with emerging networking architectures. We expect this review will provide a better comprehension of the opportunities for biologically and socially inspired computing inside technological developments and encourage other researchers to explore these approaches as part of their future work.

3. Fairness and computational justice: related work

3.1. Introduction

There is a trend towards the automation of self-organizing networks in which users can be both producers and consumers of resources. For example, in ad hoc networks, energy grids, and sensor networks, there is no centralized controller or other orchestration forms. Instead, the system requires cooperation mechanisms to collectivize resources and reach a social agreement for using them [14, 5]. This trend is imposing new conditions to deal with the distribution problem in self-organizing environments. Since users contribute with their resources, they should be able to use the resources contributed by others for their benefit. Therefore, the allocation method needs to consider not only the requirements from the system perspective but also the individual goals of each participant. This process is too fast, frequent, and complex to be performed by a human operator, requiring a computational approach to face it in the context of self-organizing networks.

Consequently, fairness and computational justice arise as an interdisciplinary research area, which is usually related to the allocation of resources. In particular, these models are responsible for distributing the benefits and charges of social cooperation in artificial systems. Although the notion of justice is frequently used in our normal activities, it is difficult to reach a universally accepted definition. This concept is subjective and may be dependent on the context of a social group. Moreover, the analysis of the distribution problem based on the tragedy of the common leads us to the question of how to manage common resources to avoid social dilemmas. Any solution requires an understanding of which processes drive human cooperation and how institutions and norms can be used to promote positive behaviors in a social system. In this chapter, we overview the current approaches and applications of computational justice. We aim to identify the potential use of these models in the context of ad hoc network through a qualitative comparison of the different methods.

The rest of this chapter is organized as follows: Section 3.2 describe the different cooperation pattern and their relationship with common-pool resources. In Section 3.3, a detail

description of the distribution problem in the context of self-organized networks is presented. We mention the particular requirements of an allocation method in environments composed of autonomous agents in which there is no central controller or other orchestrations form. Section 3.4 overview the current approaches of fairness and computational justice and present a qualitative comparison of them. We analyze models based on classical optimization techniques and socially inspired computing. Finally, Section 3.6, conclude this chapter.

3.2. Cooperation patterns and common-pool resources

Cooperation is needed to produce new forms of social organization. It can be defined as a voluntary association in which every participant can freely decide about its actions and resources to obtain individual and collective benefits. Since users contribute to their resources such as energy, memory, and processing for the network benefit, they should be able to use the resources contributed by others to achieve their own goals. Indeed, it is essential to encourage the members of the system to participate in networking tasks because higher cooperation means a better network performance [135]. It is possible to classify cooperation patterns regarding the operating condition, the number of users, and the level of reciprocity in the system. Figure 3-1 show the possible cooperation patterns in the context of self-organizing systems. These scenarios could be more complicated if we use a combination of them; however, in this research, we analyze each case independently.

In personal networks, all computational resources are configured to accomplish the goals of a single user. The cooperation pattern is trivial because the network responds to a single agent, turning the optimization of resources into the most critical task. In private networks, there is an implicit relation of reciprocity, and all interactions are based on altruistic behaviors. Either in personal and private networks, the participants are willing to share their resources, and they do not need additional mechanisms for achieving cooperation. In contrast, public networks are composed of a set of unknown users who do not have a previous trust relationship. Thus, they require to define a mutually agreed set of rules to promote cooperation among them. In this work, we analyze cooperation patterns in the context of public networks due to their applicability in ad hoc networks and mobile clouds.

Accordingly, cooperation patterns in public networks can be divided into two categories: credit-based models and trust models. The first one is based on an economic incentive to promote interaction among network components. In such models, networking tasks are treated as services that can be charged to users through virtual currencies. Some representative proposals of this approach are presented in [136, 137]. Similarly, models based on trust and reputation can also deal with selfish behaviors. If a node does not cooperate with the network,

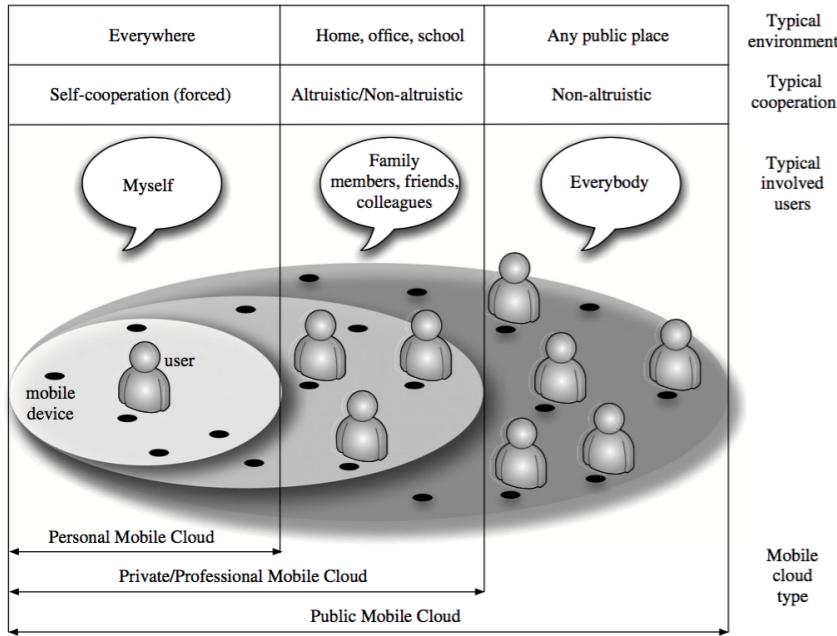


Figure 3-1.: Cooperation forms in self-organizing open systems [61].

the assumed nodes, reciprocally, may deny cooperation in future interactions. The trust level is dynamic and evolves according to network changes. This behavior creates groups of users who can exploit computational resources through distributed computing [61, 4, 3]. The conditions required to achieve cooperation in self-organized systems have been widely studied by game theory, in which models of conflict and cooperation between rational decision-makers are formally analyzed.

It is essential to mention that cooperation may emerge even in scenarios in which agents do not have an initial cooperative strategy. Therefore, we need to analyze the conditions in which a game may become cooperative, unviable, or unprofitable [138]. For example, Tit for Tat (TFT) provides a well-known framework to achieve emergent cooperation based on the past behavior of other players. Nevertheless, even TFT may be defeated whether a large population of selfish nodes appears, or due to failures in message exchange [139, 140]. Furthermore, cooperation patterns of living systems (biological, social, political, and economic) have been analyzed for disciplines like philosophy, social science, and mathematics to inspire technological solutions for artificial systems. As part of the process of designing self-organizing open systems, it is needed to establish a set of rules that allow us to regulate the benefits of cooperation, in particular, the ones related to the distribution of common-pool resources [4, 3].

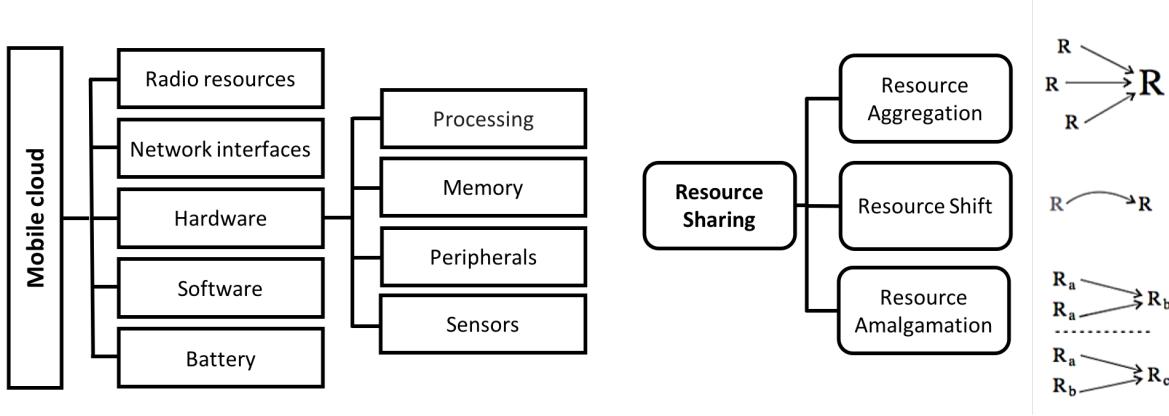


Figure 3-2.: Available resources for mobile clouds [61].

3.3. The distribution problem in self-organizing networks

To explain the significance of the distribution problem in technological developments and engineering, we will use a simple scenario in the context of the ad hoc networks and mobile clouds. One advantage offered by these technologies is the opportunity to share resources among nodes and users through the combination of wireless technologies, mobile devices, and the network capacity to operate as a self-organizing system. A mobile cloud is a cooperative arrangement of dynamically connected nodes sharing opportunistically computational resources. Energy, storage, communication interfaces, and software can be exchanged, moved, augmented, and combined in novel ways [61]. These applications arise due to the capability of the nodes to cooperate and share resources aiming to achieve a specific goal; resources like speakers, CPU, cameras, display, sensors, storage, and energy can be part of cooperation processes and used for achieving individual and collective purposes. Figure 3-2 shows the computational resources that can be used as part of the cooperation process and different approaches for combining them.

In such scenarios, all agents (nodes, users, or services) have some conception of value that represents what resources they require for achieving their goals; for instance, sunlight, information, CPU, memory, or energy. These requirements create a value system that describes any structure or process that agents can use to make choices and act to improve their status according to the system state. Given these conditions, it is necessary to consider two levels of analysis to understand the distribution problem: in the first place, the system level (or a macro-level) wherein all agents are all combined and can cooperate for achieving an overall successful outcome. At this level, we find properties like sustainability, system functionalities, fairness, or any measure that allows analyzing the system as a single unit. In the second

place, we have the agents level (or micro-level) wherein autonomous agents are pursuing their agendas according to a self-cost-benefit analysis; the agents' strategies, goals, actions, and individual behaviors are found at this level.

For studying the distribution problem, it is essential to consider both levels of analysis because of the rules and dynamics that govern the whole (the system level), and those that govern the parts (the agent level) are not aligned. This condition creates a dynamic based on cooperation and competition in which individual rationality leads to collective irrationality, i.e., when the individually rational behavior leads to a situation in which everyone is worse off. These circumstances are known as social dilemmas. There are many scenarios in which agents need to face a situation of defecting or nor not each other in the presence of common goals [133]. Likewise, a group of agents facing a social dilemma may thoroughly understand the situation, may appreciate how each of their actions contributes to a negative outcome, and still be unable to do anything to change the result. There is a considerable part of rational theory related to social dilemmas, and we need to consider these aspects in order to propose solutions for solving distribution problems in self-organizing open systems.

Furthermore, artificial systems usually use optimization techniques for improving the usage of resources and maximize the system performance regarding energy consumption, processing, availability, or other global properties (only considering the system level). This approach is useful if the participants depend on the resources, and they do not have another alternative but to be part of the system. For example, this is a typical scenario in applications in which a centralized controller or an external entity allocates the resources trying to maximize the system performance without recognizing the individual goals of the agents involved in the process. Nevertheless, in systems that need the voluntary participation of autonomous agents and their provision of resources, this approach cannot ensure the sustainability of the cooperation processes and probably lead the system to a social dilemma [133]. As a result, the distribution problem in self-organizing open systems cannot be solved through classic optimization techniques and needs to consider aspects like the agents' preferences, the cooperation mechanisms, the long-term sustainability of the system itself, and the interdependence between the two levels of analysis mentioned above [141].

In this regard, the distribution problem can be defined as follows: let be R a common and divisible resource that needs to be allocated to a set of n agents, in which each agent i has a demand $d_i \leq R$ for some portion of the resource. The resource is non-excludable and rivalrous, i.e., it is impossible or costly to prevent an agent from using the resource, and when someone uses the resource, it reduces its availability for others. In other words, given a vector of demands $\langle d_1, d_2, \dots, d_n \rangle$ and a divisible resource R , the distribution problem can

be seen as the process in which a set of agents determine by themselves an allocation vector $\langle r_1, r_2, \dots, r_n \rangle$ in which r_i represents the amount of R allocated to the agent i . $\sum r_i \leq R$

It is essential to mention that this process is made in the absence of a centralized controller or other orchestration forms; there is no information about the global state of the system, and it is difficult to predict changes in the operating conditions. The system is composed of a set of interdependent and heterogeneous agents of different provenance, with possibly conflicting goals, interacting asynchronously, in parallel, and peer-to-peer in a dynamic environment. Moreover, there is a scarcity condition, i.e., there are enough resources to keep the agents satisfied in the long-term but insufficient resources to satisfy everyone's demands at the same time [60, 65]. It is possible to explore solutions for this problem, defining a mutually agreed and understood set of rules to perform the allocation of resources, and helps the system to find equilibrium between the two levels of analysis mentioned above. However, before exploring any solution, it is necessary to consider some questions:

– *How to define an allocation method?* There are many ways in which it is possible to map the agent's demands to an allocation vector. For example, a dictatorial solution may satisfy the needs of the dictator and divide what is left among the remaining agents. A solution based on merits may distribute the resources according to the social value of each agent. A competitive solution may use auctions. Other schemes may allocate the same amount to each agent without considering their particular demands. However, beyond any solution, it is essential to analyze the distribution problem in the context of the self-organizing open systems; in particular, we are interested in both promoting cooperation and ensuring the sustainability of the system. Furthermore, the presumption that a Leviathan is necessary to elude social tragedies leads to the wrong idea that centralized control is the only way to manage common-pool resources. In Section 4.2 a proposal based on self-organizing institutions that may deal with these requirements is presented.

– *Is the allocation fair?* If the agents perceive the distribution of resources as a fair process, they are likely to comply with the rules of the system, cooperate with others, and not turn into free-riders. The aim is keeping them as satisfied as possible to reinforce positive behaviors. In this case, fairness is used to evaluate the allocation of resources in terms of the distribution and the individual satisfaction of the participants. Nevertheless, the essential question remains: how to achieve a fair distribution among a set of autonomous agents? The answer to this question varies among the fields (philosophy, economics, politics, computer science), and there is no single solution. Besides, most of the proposed methods in the context of computational justice allow evaluating the allocation of resources, but they do not give a method for achieving a fair distribution.

–*Is the method sustainable over time?* The majority of the methods available for allocation of resources in artificial systems do not consider temporal aspects; they are used to analyze static problems in which it is necessary to distribute resources among a set of agents in a given instant of time. These methods do not take into account a sequence of allocation problems that involve the same agents and possible changes in the environmental conditions. Since we are interested in long-term properties like sustainability and enduring cooperation processes, we require to consider the potential long-term consequences of an allocations scheme. For example, it is needed to distinguish between an outcome that seems to be unfair but proves to be fair over time and just a simple sequence of unfair outcomes.

–*How to design cooperation and distribution rules that lead to good social outcomes?* Since we are interested in self-organized systems in which agents have the autonomy to take part or not in the allocation process, it is necessary to create mechanisms to aggregate and combine different agents' preferences in a sensible way. In other words, it is essentially a problem of voting. These mechanisms allow the participants to decide as a group and considering the interest of all agents before making any decision that affects the whole system. Furthermore, these methods create a relationship between the two levels of analysis mentioned before; the idea is to include the agents' preferences in the choices related to the distributions process to promote cooperation and increase the individual satisfaction of the agents.

–*Is the method scalable?* According to Olson [142], unless the number of individuals is quite small, or unless there is coercion or some control mechanism to make individuals work in their common interest, rational self-interested agents will not work to achieve collective goals. An agent that cannot be excluded from obtaining the benefits of a collective good once the good is produced has a little or any interest to contribute voluntarily to the provision of that good (free-rider). However, Olson considers as an open question whether intermediate-size groups will or will not voluntarily provide collective benefits. His definition of an intermediate-size group depends not on the number of agents involved in the process, but on how noticeable each agent's actions are. Besides, the information related to the system state could be socially spread but not useful; it needs to be absorbed by agents. Thus, the limit in the capacity of an agent to make decisions properly depends on its ability to obtain and process information.

–*Where the resources come from?* To understand where the resources come from, we can consider two different scenarios. On the one hand, common but finite resources available in the environment owned by the members of a community that can be used without restriction. For instance, grid computing provides a distributed computing infrastructure based on common resources, in which agents (users or services) take advantage of idle computing

resources to process massive, long-running, computationally-intensive tasks. This scenario is usually analyzed through the Prisoner's Dilemma. On the other hand, some resources arise as a result of the cooperation processes; they appear because of the combination of the individual contribution of the members of the system. For example, in mobile clouds, each agent provides a part of its resources to create common-pool resources to accomplish individual and collective goals.

3.4. Fairness and computational Justice: background

Justice is an interdisciplinary research area that studies different techniques for distributing the benefits and charges of social cooperation. These ideas have been used to design algorithms to deal with distribution problems in communication networks and also analyze properties like efficiency, load balancing, and QoS. For instance, in a computational system, the resources expect to be distributed equitably amongst all processes and their threads. In computer networks, all nodes expect to have a fair amount of the available bandwidth to send and receive data. Although the notion of Justice initially belongs to social science and philosophy, nowadays, it is considered as an inspiration source for solving computational problems related to the allocation of resources in self-organizing open systems. Table 3-1 review the different applications of these models according to the OSI model.

In order to overview the fairness and computational Justice models in the context of communication systems, we need to classify them into two categories. First, those that have been developed in the context of engineering through classical optimization techniques; this approach mainly takes place in the context of distributed systems and centralized networks. Second, computational justice models inspired by social theories like Distributive Justice and Institutions in which characteristics like cooperation, autonomy, and self-organization are essential aspects of the system operation. Although this research focuses on socially inspired computing, the following Sections reviewed both approaches for the sake of clarity. We do not aim to present a detail description of each model but to analyze their main characteristics for comparison purposes.

3.4.1. Classical engineering approach

Different fairness approaches have been proposed during the last years to assess the allocation of resources in communication systems. In this case, we assume that there is a divisible and finite resource whose total amount is R , and there are N individuals sharing this resource through an allocation vector $\mathbf{A} = \langle r_1, r_2, \dots, r_n \rangle$ in which r_i represents the amount of the resource R allocated to the agent i and $\sum_{i=1}^N r_i \leq R$. According to their measurability,

Table 3-1.: Application of computational justice in communication systems

Layers	Resource allocation	Performance measurements
Application	Computational resources, distributed operation of resources. Mobile clouds.	Quality of service (QoS) related to application performance. Fairness regarding access to resources over time.
Transport	Traffic flow control in the transport layer, use of ports, and data buffer.	(QoS) on aspects such as congestion control, loss-rate, reliability, end-to-end communications.
Network	Routing, route discovery, message delivery and congestion control at the network layer	Load balancing routing and security.
MAC	Access to multiple channels, collision management, MAC layer flow control, Simultaneous protocol management (Wifi, Bluetooth, Zigbee)	Bit error rate (BER), interference and transmission power control
Physical	Power distribution on antennas and battery energy allocation	Lifetime, energy consumption, energy allocation

these techniques can be classified in quantitative and qualitative. A common approach is to represent quantitative measures through real values using a fairness function $f : \mathbb{R}^n \rightarrow \mathbb{R}^+$ based on the allocation vector \mathbf{A} in which n represent the number of agents involved in the process; this function is expected to satisfy the following conditions:

- R1: $f(\mathbf{A})$ should be continuous on $A \in \mathbb{R}^n$.
- R2: $f(\mathbf{A})$ should be independent of the number of participants N .
- R3: $f(\mathbf{A})$ should be easily mapped on to $[0, 1]$.
- R4: $f(\mathbf{A})$ should be easily extendable to multi-resources case.
- R5: $f(\mathbf{A})$ should be sensitive enough to variations in A .

The requirements R1 and R2 imply that the fairness function is independent of the number of participants and exist in every possible distribution of resources. The third requirement R3 is related to the applicability of the fairness function and its capacity to give an intuitive and direct impression about the current resources allocation. The requirement R4 aims to make the fairness function realistic and implementable in different computational applications. Finally, requirement R5 ensures the fairness function is sensitive enough to cover small variations in the allocation of resources.

Among the most used quantitative measures, we found the Jain Index. It provides a fairness perspective from the system level. Even though this is one of the most used approaches, it is not able to identify unfairly treated individuals and also requires complete information of the system to operate. Tian Lan's measure is another quantitative model that provides a general approach for fairness analysis. It can be transformed into the Jain's index, Entropy, max-min, and proportional fairness. Although this is a more general approach, it still requires complete information to perform a fairness evaluation. Entropy can also be used

Table 3-2.: A comparison of computational justice model in the context of engineering

Models	Control	Priority	Information	Measure	Temporality	Complexity	Year
Jain's index	Centralized	No	Complete	Quantitative	No	Simple	1984
Envy-based	Either	No	Either	Quantitative	No	Complex	1995
Proportional	Centralized	Yes	Complete	Qualitative	No	Medium	2004
Max-min	Either	Yes	Either	Qualitative	No	Simple	2007
Tian Lan's	Centralized	No	Complete	Quantitative	No	Complex	2010
Entropy	Centralized	No	Complete	Quantitative	No	Simple	2011
Altman	Centralized	Yes	Complete	Qualitative	Si	Medium	2012
Nowicki	Centralized	No	Complete	Quantitative	No	Medium	2016

as a quantitative measure, but it can deal with neither small variations in the resources allocation nor locates unfairness in the system. A detail description of these measures and other quantitative methods can be found in [122].

On the other hand, qualitative approaches cannot assess the allocation of resources in terms of a real number. They evaluate if the distribution method can achieve fairness and provide some guidance for improving the allocation process regarding a well-defined set of conditions. Two of the most conventional approaches are max-min and proportional fairness. The first one represents scenarios in which a system cannot increase any user allocation without decreasing the resources of others. Max-min fairness has applications in flow control, bandwidth sharing, and radio channel accessing. Proportional fairness considers a system as fair if it satisfies a set of conditions related to the difference among the users' allocation and the availability of resources in the system. Several approaches for proportional fairness have been developed and can be easily extended to multi-resources scenarios. A detail description of these models and other qualitative methods can be found in [122].

In general terms, the classical engineering approach uses objective functions to optimize or evaluate the allocation of resources regarding a set of parameters. However, although there is a diversity of models, they do not provide a well-defined method to achieve a fair distribution process. Table 3-2 presents a comparison of the most used techniques in the context of engineering. We consider the following criteria: *Control* to indicate whether the model was developed for centralized or self-organizing systems. *Priority* as the possibility of assigning individual weights during the distribution process. *Information* to show if the model requires complete information of the system to operate. *Measurability* to specify if the measure is quantitative or qualitative. *Temporality* if the model considers changes in the allocation of resources over time. *Complexity* to describe the computational cost of the fairness evaluation; three levels simple, medium, and complex are used according to the analysis presented

Table 3-3.: A comparison of socially inspired computational justice model

Models	Based on	Control	Priority	Applied	Measure	Temporality	Year
J. Pitt	Ostrom - Rescher	Distributed	Yes	General model	Quantitative	Yes	2012
J. Pitt	Ostrom - Rescher	Distributed	Yes	MAS	Quantitative	Yes	2014
D. Busquets	Ostrom - Rescher	Distributed	Yes	Smart grids	Quantitative	Yes	2015
F. Torrent	Ostrom - Rescher	Distributed	Yes	Smart grids	Quantitative	Yes	2016
J. Garbisso	Ostrom - Rescher	Distributed	Yes	VANETs	Quantitative	Yes	2017

in [122]. Note that none of the mentioned measures perform properly regarding all criteria; however, it is possible to use a combination of them as an alternative solution.

It is essential to mention that the majority of the proposals presented in Table **3-2**, were developed for centralized systems. Although the max-min model can operate in self-organizing networks, it is not effective in dynamic and uncertain environments. Additionally, most of the models need complete information to compute the fairness measure, and it is not possible to guarantee that condition in a self-organizing environment. Even though the model proposed by Altman takes into account the time in the allocation of resources, it was designed for a centralized operation. Consequently, these approaches are not suitable for facing the distribution problem in self-organizing open systems in which distributed control, incomplete information, and uncertain environments are intrinsic aspects of the network operation.

3.4.2. Socially inspired computing approach

Social inspired computing aims to use social behaviors to create computational algorithms that can deal with the decentralized nature of self-organizing systems. In this regard, computational justice uses cooperation patterns, the idea of social Institutions, and Distributive Justice theories to provide an allocation method in which a set of autonomous agents can share and distribute their resources without the intervention of a central controller. Table **3-3** presents an overview of these models. One of the most remarkable proposals in this context is the work presented by Pitt et al. [1] in which a general method for allocating resources in self-organizing open systems is developed. This proposal has been used for multiple applications in smart-grids, multi-agent systems, and the idea behind of democratizing technological platforms. It uses the Borda voting system to combine the agent preferences and establish a priority list that allows the Institution to perform the allocation of resources according to the canons of justice. Although this approach presents an effective method to face the distribution problem, it is computationally expensive due to the voting rounds performed during the process.

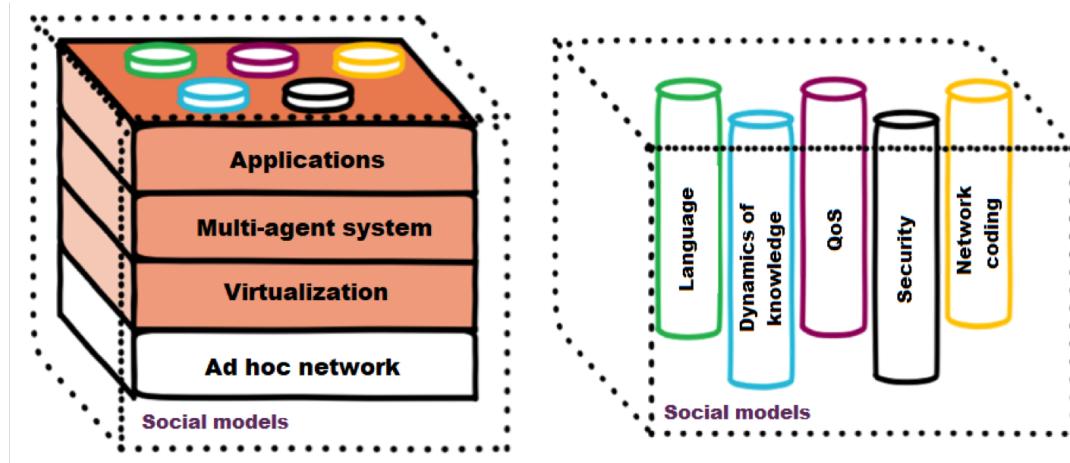


Figure 3-3.: TLÖN: a socially inspired computing system

Socially inspired computing arises as a potential solution to face the distribution problem in the context of self-organizing systems. These models are developed for scenarios in which attributes like incomplete information, decentralized operation, and autonomous behaviors are a fundamental part of the system operation. It is important to remark that there are no applications of these models in the context of computer networks; however, the current proposals provide us a set of criteria and principles for developing a solution for communication systems. In chapters 4 and 5, we present a proposal for dealing with the distribution problem in self-organizing systems, in particular, for ad hoc networks.

3.5. TLÖN project: a socially inspired computing system

The TLÖN Research Group at the National University of Colombia is developing a project which aims to design and implement a socially inspired computing system using ad hoc networks as the base technology. The idea is to create computational models based on the behavior of social and biological systems that allow us to control the decentralized behavior of an ad hoc network. In this regard, one of the expected functionalities of the TLÖN System is to provide a platform that allows users to share and distribute their resources freely without a central controller. Thus, a computational model for dealing with the distribution problem in self-organizing systems is required. In Figure 3-3, the general aspects of the TLÖN project are presented.

- *Ad hoc network:* it is a self-organizing network in which there is no pre-existing infrastructure; nodes are autonomous, and they can decide what will be their behavior in

the system. This layer provides the computational infrastructure and resources in the system.

- *Virtualization*: it is responsible for operating, collecting, and implement the distributed operation of resources. The aim is to provide applications like Mobile Clouds in which the system can use the resources of the whole system for specific purposes.
- *Multi-agent system*: this layer is responsible for managing and controlling the services offered by the system. We use the agent paradigm to model the system components as autonomous agents; the behavior of the whole system is based on cooperation and competition. The idea is to create an internal social organization based on the notions of community, institutions, and justice.
- *Application*: it represents the set of solutions that can be modeled and implemented through social inspired computing for ad hoc networks. In this case, applications for sensor networks, mobile clouds, and IoT applications can be considered.
- *Programming language*: in order to implement socially inspired computing applications, a specific domain programming language is needed to control and configure the system.
- *Knowledge Dynamics*: it refers to the available mechanisms of knowledge representation and beliefs inside the TLÖN system.

This research is part of the design and development process of the TLÖN System and is directly related to the multi-agent system layer. We aim to model and implement the required computational mechanisms to allow the system to deal with the distribution problem in the context of ad hoc networks.

3.6. Conclusions

In this chapter, the distribution problem, cooperation patterns, and some application of common-pool resources were introduced to evaluate their applicability in the context of self-organizing networks. We overview and compare the current fairness and computational justice models. Two different approaches were presented: first, the classical engineering approach, which usually uses a set of measures (quantitative or qualitative) to assess a resource allocation according to a set of parameters. Although there are a considerable number of proposals related to this approach, most of them need a centralized operation and complete information of the system to provide a fairness analysis. Second, socially inspired computing approach which uses social concepts like institutions, distributive justice, and cooperation to create allocation methods based on the behavior of human society. Despite there is no direct application of socially inspiring computing in communication systems, they provide a set of principles that can be used to face the distribution problem in self-organizing communication networks; in particular, ad hoc networks.

4. A computational justice model for resources distribution in ad hoc networks

4.1. Introduction

As we mentioned in chapter 3, the distribution problem in the context of self-organizing networks can be defined as follows: let be R a common and divisible resource that needs to be allocated to a set of n agents, in which each agent i has a demand $d_i \leq R$ for some portion of the resource. In other words, given a vector of demands $\langle d_1, d_2, \dots, d_n \rangle$ and a divisible resource R , the distribution problem can be seen as the process in which a set of agents determine by themselves an allocation vector $\langle r_1, r_2, \dots, r_n \rangle$ where r_i represents the amount of resources R allocated to the agent i . This process is made in the absence of a centralized controller or other orchestration forms; there is no information about the global state of the system, and it is difficult to predict changes in the operating conditions. Also, the system is composed of a set of heterogeneous agents with possibly conflicting goals, interacting asynchronously, in parallel, and peer-to-peer in an uncertain environment [60, 65].

In this chapter, we argue that it is possible to deal with distribution problems in self-organizing communication networks using a combination of socially inspired computing and agent-based modeling. We intend to use living systems analogies to create computational mechanisms inspired by the behavior of biological and social systems. This proposal is comprised of: (a) the Ostrom's principles for enduring institutions; (b) a cooperation model based on stochastic games; (c) the Rescher's approach of distributive justice; and (d) adaptation mechanisms through voting systems and genetic algorithms. We take as reference the model introduced in [1]. Essentially, the similarities between the previous works and ours are the Ostrom's approach for institutional analysis and the use of the Rescher theory of distributive justice. However, the cooperation patterns based on stochastic games and the use of genetic algorithms as an adaptation mechanism are totally different. Also, we used a variation of the equality canon of justice that allows us to improve the response of our proposal to selfish behaviors in the system.

Accordingly, this chapter is organized as follows: Section 4.2 reviews the Ostrom's approach for self-organizing institutions. Section 4.3 presents cooperation patterns based on stochastic games as an abstract representation of the distribution process in self-organizing networks. Section 4.4 shows the canons of justice based on the idea of legitimate claims. Section 4.5 presents adaptive strategies based on genetic algorithms and the Borda voting method; the idea is to allow the institutions to adjust their parameters to unexpected changes in the environment. Section 4.6 combine all these elements as a set of exchangeable parameters in the internal architecture of a multi-agent system. Section presents 4.7.3 the experimental anSection 4.8 concludes this chapter.

4.2. Self-organizing institutions for collective actions

Based on extensive work on self-organizing institutions, Ostrom argues that a set of self-interested agents sharing common-pool resources not always will end in a social dilemma like the tragedy of the commons [143]. She noted that in the context of human societies, these problems have often been solved through the use of institutions combined with an understanding of which processes drive human cooperation and how norms and other feedback mechanisms can be used to reinforce positive behaviors. Accordingly, Ostrom identified eight design principles as necessary and sufficient conditions for common-pool resources managed by self-organizing institutions to endure. These principles are listed in Table 4-1. We are interested in the principles 1,2 and 3, in order to design institutions to manage common-pool resources in the context of ad hoc networks for situations in which the optimal distribution of resources is less important than the persistence of the resources available in the environment.

Institutions are responsible for establishing the structure and regulating the behaviors in a social system; they determine who is eligible to make decisions, what actions are allowed or constrained, and contain prescriptions that forbid, permit or require some actions or outcomes according to a specific context. Using the Ostrom's approach, we can describe an institution through three different levels of nested rules: first, the *operational-choice rules*, which are concerned with the actions that affect the physical world like resources appropriation, provision, monitoring, and enforcement. Second, the *collective-choice rules*, which are concerned with selecting the operational rules, aggregate the agents' preferences, dispute resolution, and policy-making. Third, the *constitutional-choice rules* which are responsible for determining who is eligible, and what specific rules are to be used, to modify the *collective-choice rules*. The three levels are role-based, mutually agreed, mutable and nested in action situations.

Table 4-1.: Ostrom's design principles

-
- 1. Clearly defined boundaries.**
Individuals who have access rights to the institution must be clearly defined, as must the boundaries of the institution itself.
 - 2. Congruence between institutional rules and local conditions.**
There should be a congruence between appropriation and provision rules and the state of the prevailing environment.
 - 3. Collective-choice arrangements.**
Most individuals affected by the rules of the institution must be able to participate in the selection and modification of those rules.
 - 4. Monitoring.**
Monitors who actively audit the application of the institutional rules must be, or be accountable to, those affected by the institutional rules.
 - 5. Graduated sanctions.**
Members who violate institutional rules are sanctioned according to the seriousness of the violation. These sanctions are given by the other members of the institution.
 - 6. Conflict-resolution mechanisms.**
The members of the institution should have rapid access to fast, cheap conflict-resolution mechanisms..
 - 7. Minimal recognition of rights to organize.**
Existence of and control over their own institutions; they are not challenged by external authorities.
 - 8. Systems of systems.**
The rules governing the institution and the application of the previous seven principles are organized in multiple levels of nested enterprises.
-

Note that the idea of institutions proposed by Ostrom does not say anything about the fairness of the distribution method or how it could affect the environment state or the cooperation in the system. In fact, the notions of fairness and cooperation are highly related to the allocation method; if the agents perceive the distribution of resources as a fair process, they are likely to comply with the rules of the system and may be willing to cooperate in future interactions. In this regard, we want to complement the Ostrom's approach for enduring institutions with the theory of distributive justice to evaluate if an allocation scheme can be considered as a fair process. Our goal is not only design institutions for the enduring common-pool resources but also use the idea of justice for promoting cooperation among the members of the system.

4.3. Cooperation in stochastic games

Social dilemmas like the tragedy of the commons lead to analyze how to manage public goods in environments composed of autonomous agents in which there is no centralized control or information. Game theory suggests that iterated games based on reciprocity and trust can deal with these dilemmas. However, the majority of the previous works assume that the public good remains constant over time, no matter the outcome of previous interactions. In this Section, we introduce a cooperation framework based on stochastic games, which allow us to capture the idea that agents affect and are affected by the value of the public good [144]. The agents' decision of whether to cooperate or to defect not only affects their payoffs but also the game they will play in the next round. Thus, a group of agents can find themselves in one of the multiple states in which they interact in a social dilemma with different payoffs, capturing how the current conditions of the physical and social environment affect the feasible outcomes of the agents.

Consequently, we can use the prisoner's dilemma and the linear public good game in their stochastic versions for modeling cooperation patterns in self-organizing systems, in particular, ad hoc networks and mobile clouds. On the one hand, the prisoner's dilemma allows us to represent common but finite resources available in the environment owned by the members of a community that can be used without restriction. On the other hand, the linear public good game allows us to represent resources that arise due to the cooperation among a group of agents. These two scenarios are examples of social dilemmas [133]. In both cases cooperation entails a cost $c > 0$. In the prisoner's dilemma cooperation produce a benefit $b_i > c$ to the co-player where b_i depends on the current state i . In the linear public good game the aggregated cost is multiplied by a factor r_i (with $1 < r_i < n$) and allocated among all agents. Game 1 is more profitable than game 2 if $b_1 > b_2$ or $r_1 > r_2$ and the agents only will play game 1 if everyone has cooperated in the previous round. Figure 4-1 describes these scenarios.

It is important to mention that in stochastic games, cooperation evolves because defectors are affected in two different forms: first, they could receive less cooperation of other agents in the next rounds if they do not comply with the system's rules. Second, the agents move collectively towards a less profitable game when some of them are defecting. This situation can be seen as a reduction of the resources available in the environment because of the presence of free-riders in the system. Likewise, if everyone cooperates, the environment could recover, and the original value of the public good can be restored. The different states of the environment correspond to different games that will be played according to the collective behavior of the agents: the more cooperative the agents, the more resources will be.

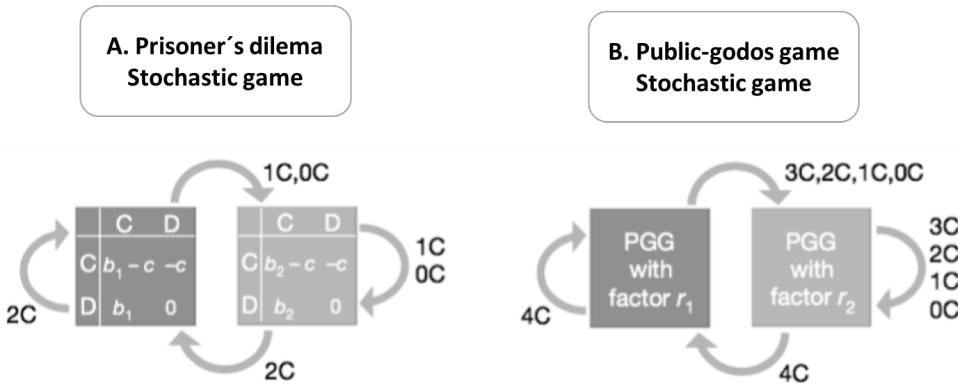


Figure 4-1.: Stochastic games [144].

We can define a stochastic game using five objects [144]:

1. The set of players N ;
2. The set of possible states S ;
3. The set of actions $A(s_i)$ that are available to each player in a the state $s_i \in S$;
4. The transition function Q that describes how the current state and the agents' actions determine the next state; and
5. A payoff function u that describes how the payoff of the players depends on the agent's actions and the current state.

In this case, the model does not specify how much time passes between consecutive rounds nor restrict the payoffs that are available in each round. We consider stochastic games in which agents can choose between cooperate or defect and their action set is defined by $\{C, D\}$. Initially, agents are in state 1, and the payoffs per round are symmetric and change according to a discounted value during all rounds in the game. In the following Sections, the prisoner's dilemma and the linear public good game are presented.

4.3.1. The prisoner's dilemma

The prisoner's dilemma has been extensively used in game theory to explore situations in which autonomous agents act pursuing their interest in environments in which they are not obligated to cooperate even when it seems to be the best for all the members of the system. This model has been usually used to formalized the tragedy of the commons and symbolize the degeneration of the environment to be expected in situations in which many agents use a common scare resource. The prisoner's dilemma is a non-cooperative game in which each agent has a dominant strategy (to defect no matter what the other agents choose) and represents a paradox in which the best individual strategy leads a situation in which

everyone is worse off. The game is played in consecutive rounds, and the agents take their decisions under self-interest analysis. In each round, each agent i :

- Determines its needs of resources, $q_i \in [0, 1]$
- Makes a demand for resources, $d_i \in [0, 1]$
- Receives an allocation of resources, $r_i \in [0, 1]$
- Makes an appropriation of resources, $r'_i \in [0, 1]$

The total amount of resources owned by an agent at the end of the round is given by the resources appropriated rather than allocated. The agents freely decide the amount of resources they want to take from the public good independently of the allocation they received. In this game, cooperate means the agent makes an appropriation lower or equal than the allocation assigned by the institution. Also, in order to include a direct relationship between the amount of resources and the agents' utility, the payoff function u is given by the following matrix:

		C	D
C	$(b_1 - c)R_t$	$-cR_t$	
D	$b_1 R_t$	0	

in which, $R_t \in [0, 1]$ is a value that represents the proportion of the available resources in the environment. After each round, the resources are multiplied by a coefficient k that embodies the interdependence between the collective behaviors of the agents and the availability of resources in the system; if all the agents cooperate, the resources will increase or at least remains in the same value. Otherwise, there will be fewer resources in the next rounds. Note that the value of the coefficient k represents both the capacity of the environment to recover itself under proper conditions and a degeneration process as a result of free-riders in the system.

4.3.2. The linear public goods game

The problem of the voluntary provision of resources has been usually analyzed using the linear public good game [145]. Nevertheless, due to the nature of open self-organizing systems, this model presents some limitations that need to be considered before applying it in the context of ad hoc networks. For example, it assumes that the public payoff is equally distributed even when it is possible for the appropriation to exceed allocation; that there is a full disclosure of all information required for the process; that there is no cost related to monitoring; that the utility for all resources are the same no matter if they are needed or not. As a consequence,

in order to get a more realistic model, we relax some of these conditions using a variation of this game [146]. In this case, n agents form a cluster in which each agent i owns a quantity of some divisible resource and freely decides if contribute or not to the public good. We assume that agents take their decisions under self-interest analysis, and the game is played in consecutive rounds. In each round, each agent i :

- Determines the resources it has available, $g_i \in [0, 1]$
- Determines its needs of resources, $q_i \in [0, 1]$
- Makes a demand for resources, $d_i \in [0, 1]$
- Makes a provision of resources, $p_i \in [0, 1]$ ($p_i \leq g_i$)
- Receives an allocation of resources, $r_i \in [0, 1]$
- Makes an appropriation of resources, $r'_i \in [0, 1]$

The total amount of resources owned by an agent at the end of the round is given by $R_i = r'_i + (g_i - p_i)$, in which R_i is the sum of resources appropriated by the agent and the ones that it keeps for itself. The contributions of all participants are summed, and the payoff u_i for the agent i is given by:

$$u_i = \begin{cases} a(q_i) + b(R_i - q_i) & \text{if } R_i \geq q_i \\ a(R_i) - c(q_i - R_i) & \text{otherwise,} \end{cases} \quad (4-1)$$

where a, b and c are coefficients in \mathbb{R} that represent the relative utility of getting the resources that are needed, getting resources that are not needed, and not getting the resources that are needed.

Furthermore, independent of its utility and the cooperation pattern (the prisoner's dilemma or the linear public good game) each agent i makes a subjective assessment of its satisfaction S_i expressed as a value in $[0, 1]$ according to the relationship between its allocation and its demands. In this regard, we can define the satisfaction level of the agent i in the round $t + 1$ as follows:

$$S_i(t + 1) = \begin{cases} S_i(t) + \alpha [1 - S_i(t)] & \text{if } r_i \geq d_i \\ S_i(t) - \beta(q_i - R_i) & \text{otherwise,} \end{cases} \quad (4-2)$$

where α and β are coefficients in \mathbb{R} , which determine the rate of reinforcement of satisfaction and dissatisfaction of each agent. As a result, choosing different combinations of α and β allow us to model different behaviors in the agents. For example, high values of α and low values β enable us to model agents with a high level of tolerance to situations in which they do not get what they need. On the other hand, high values of β will make the agents be dissatisfied more quickly, and therefore, they would stop following the institutional rules. This scenario is modeled through a threshold value τ and an interval value m . If for m

consecutive rounds the agent i evaluate $S_i < \tau$ as true, it will stop cooperate. In the case of the prisoner's dilemma, the agents appropriate an amount of resources greater than the allocated (they turn into free-riders). In the linear public good game the agent leaves the cluster.

4.4. Distributive justice: Rescher's approach

Distributive justice is related to the fair distribution of resources; it includes the available resources in the environment, the method responsible for the distribution, and the outcomes resulted from a specific allocation. The notion of justice is an essential characteristic of any social system because it is responsible for distributing the benefits and charges of social cooperation. We can explore the distributive justice theories through three main families: *equality and need*, *utilitarianism and welfare*, and *equity and desert*. The first family concerns about the welfare of those least advantaged; in this case, the society is organized to give an equal satisfaction of the basic needs. Egalitarianism, Rawls' theory of justice, and Marxism are examples of this approach. The second family, *utilitarianism, and welfare* focuses on an efficiency principle that tries to maximize the global welfare of the society, the greatest good for the greatest number. Utilitarian theory, Pareto principles, and Envy-freeness belong to this family. The third family, *equity, and desert* states that individuals should receive an allocation that is proportional to their contribution to society. Nozick's theory of libertarian rights belongs to this approach. A detailed description of the three families can be found in [147].

In this research, we use the idea of a pluralistic justice proposed by Konow [147] and previously applied in the methodology introduced in [60, 1, 64]. We aim to consider several perspectives during the process of determining what a fair distribution of resources is and not restrict this consideration to a specific theory. This analysis coincides with the idea of distributive justice proposed by Nicolas Rescher that states individuals should be treating according to the notion of legitimate claims and canons of justice. Table 4-2 summarizes these principles as the canons of equality, need, effort, supply and demand, productivity, and social utility. The Resecher's idea of justice requires to determine, in context, what the legitimate claims are, how they are accommodated in case of plurality, and how they are reconciled in case of conflict. We intend to combine these canons with the Ostrom's approach for institutions in order to deal with distribution problems in ad hoc networks. Thereby, we use a computational representation of each canon computing a total order over the set of agents. In this case, T denotes the number of rounds in which the agent i participate in the allocation process. Each canon is described below [1].

Table 4-2.: Rescher's canons of distributive justice

$f_1 - Equality$	Treatment as equals.
$f_2 - Need$	Treatment according to their needs.
$f_3 - Productivity$	Treatment according to their actual productive contribution.
$f_4 - Effort$	Treatment according to their efforts and sacrifices.
$f_5 - Social utility$	Treatment according to a valuation of their social-useful services.
$f_6 - Supply and Demand$	Treatment according to supply and demand.
$f_7 - Merits and achievements$	Treatment according to their ability, merit or achievements.

$f_1 - The canon of equality$: we can represent this canon in two different ways: (f_{1a}) by ranking the agents in increasing order according to their average allocation;

$$f_{1a}(i, T) = \frac{\sum_{t=0}^T r_i(t)}{T} \quad (4-3)$$

(f_{1b}) by ranking the agents in increasing order of the everyman satisfaction S_i .

$$f_{1b}(i, T) = \frac{\sum_{t=0}^T S_i(t)}{T} \quad (4-4)$$

The original model of computational justice proposed by Pitt et al. [1] considers an additional representation for this canon in which the number of rounds where the agents receive an allocation is used as an equality measure. However, we do not consider this model because even if the agents do not comply with the institution rules, it allows them to get resources from the system.

$f_2 - The canon of need$: This canon ranks agents in increasing order according to their average demands; in this case, we suppose that similar agents demand average quantity over time, then the agent most in need is the one that so far has made the least demand.

$$f_2(i, T) = \frac{\sum_{t=0}^T d_i(t)}{T} \quad (4-5)$$

$f_3 - The canon of productivity$: we can represent this claim by ranking the agents in decreasing order according to their average provision:

$$f_3(i, T) = \frac{\sum_{t=0}^T p_i(t)}{T} \quad (4-6)$$

It is possible to use several alternatives for representing this canon; for example using the total provision ($\sum_{t=0}^{T_i} p_i(t)$) or the net provision that is represented by the difference between provision and allocation.

f₄ – The canon of effort: Rank the agents in decreasing order according to the number of rounds spent as *prosumer*. This role is responsible for determining which agents are members of the institution. This canon is congruent with the first institutional principle proposed by Ostrom.

f₅ – The canon of social utility: this canon is represented by ranking the agents in decreasing order according to the number of rounds spent in the *head*. This role is responsible for the allocation of resources.

f₆ – The canon of supply and demand: This canon rank the agents in decreasing order according to this measure of compliance; in other words, those agents who follow the norms of the game, which are: first, do not withhold what is available ($p_i = g_i$); second, only demand what is needed ($d_i = q_i$); and third, only appropriate what is allocated ($r'_i = r_i$):

$$f_6(i, T) = \sum_{t=0}^T [(p_i(t) = g_i(t)) \wedge (d_i(t) = q_i(t)) \wedge (r'_i(t) = r_i(t))] \quad (4-7)$$

This canon assumes that we can monitor every agent's internal state, monitoring and reporting is perfect, and the cost of monitoring is zero. However, in practice, we only monitoring which is observable and enforceable; appropriation equals to allocation.

f₇ – The canon of merits and achievements: this canon is not appropriate in this context and is not represented.

As a final step, all canons of justice are combined using the Borda voting method to generate a priority list that allows the institution to perform the allocation of resources. In this case, each canon of justice is considered as a voter who submits a full order over all agents in the system; if there are n agents, it contributes $n - 1$ points to the highest-ranked agent, $n - 2$ points to the second-highest, and so on. The Borda score of an agent is computed from the accumulation of the points given by each vote. As a result, a priority list in descending order is produced based on the results of the voting process. Finally, the institution performs the allocation satisfying the demands of the first agent in the list, then the second one and so on. This process is repeated until there are no more resources to allocate, or all demands are satisfied.

4.5. Adaptive behaviors in self-organizing systems

The idea of adaptation in self-organizing systems is different from other disciplines like machine learning, statistics, or artificial intelligence. In general terms, these disciplines have in mind a single agent acting in an environment that could be unknown, stochastic, partially observable, and so on; it could be challenging to find an optimal strategy, but there is a well-defined notion of what an optimal strategy is. In contrast, in the context of self-organizing communication networks, we have systems composed by multiple agents in which everyone is trying to adapt their strategies and achieve their goals at the same time; when an agent adapt its behavior, it is influenced not only by the environment but also by the behavior of other agents. This condition produces a high level of interdependence among the members of the system, and it is needed to provide institutions with adaptive mechanisms that allow them to adjust their parameters to react appropriately to changes in the agents' behavior and the environmental conditions.

4.5.1. The Borda voting approach

The notion of voting in the context of distribution problems can be summarized as follow: how should an institution pool the preferences of a set of individuals in order to best reflects the wishes of the population as a whole? We can find different approaches for dealing with this issue through voting systems based on plurality, approval, pairwise elimination, and so on [148]. Nevertheless, we use the Borda voting protocol, which is described as a consensus scheme that selects what is broadly accepted rather than what it is preferred just by the majority. In this method, each voter submits a ranking over the candidates and assign them a number of points; if there are n candidates, it contributes $n - 1$ points to the highest-ranked candidate, $n - 2$ points to the second-highest, and so on. The winner (or winners) is who has the maximum amount of points at the end of the voting process.

In order to use this method as an adaptation mechanism, a weight $w_i \in [0, 1]$ is attached to each canon of justice to adjust their influence in future rounds of the game. We assume the agents will vote by the canons that give them the highest utility. Each agent is treated as a voter, and the canons as the possible candidates; the Borda points of the canon f_i in a system composed of N agents are given by:

$$\text{Borda}(f_i) = w_i \cdot \sum_{j=1}^N p(j, f_i), \quad (4-8)$$

in which $p(j, f_i)$ represents the number of Borda points assigned by the agent j to the canon

f_i . In addition, the weight w_i of each f_i is updated by:

$$w_i(t+1) = w_i(t) \cdot \frac{Borda(f_i) - AvgBorda}{TotalBorda}. \quad (4-9)$$

Note that all new weights are summed and normalized to 1. This process is repeated at the end of each round and implies changes in the canons' weights based on the combined preference of all agents. Also, this method implies two voting rounds; first, a voting round performed by the institution as part of the allocation of resources in which the canons of justices are treated as voters and the agents as the candidates. This round has the purpose of prioritizing the agents for the allocation process. In the second round, the roles are exchanged: the agents are treated as voters and the canons of justice as candidates. This round has the purpose of modifying the weight of each canon and allow the institution to adjust its parameters to changes in the agents' behavior.

4.5.2. Genetic algorithms

Genetic algorithms are heuristics techniques that are based on the evolution of living systems and are used in complex optimization problems where the solution space is unfeasible. These methods work with an initial population of individuals in which each one represents a possible solution to a specific problem. Genetic algorithms are fundamentally non-deterministic and give rise to the emergence of a new organism through recombination and mutation of the genetic code. Generally, this methods perform four steps to produce a new set of solutions: *initialization*, *evaluation*, *selection*, and *reproduction*. This process is repeated until a satisfactory solution appears. We aim to use the continuous genetic algorithm approach to explore how different combinations of weights during the allocation process. The four steps are described below [149]:

1. *Initialization*: To begin GA, a set of solutions is described through a genetic code or chromosome. Usually, the first population is randomly generated and represented within a data structure. In this case, the initial population is composed of N_{pop} chromosomes defined as a combination of weights for the canons of justice; $chromosome = [w_1, w_2, \dots, w_7]$ in which $w_i \in [0, 1]$ and $\sum_i w_i = 1$.
2. *Evaluation*: this mechanism is used to measure the capacity of each solution to solve the problem; it is made through a fitness function that represents the level of optimality of each individual. Therefore, the fitness of a chromosome is defined by $f(w_1, w_2, \dots, w_7)$, in which $f(\cdot)$ represents the average satisfaction of all agents after n rounds of the game.
3. *Selection*: the genetic algorithm picks individuals out among the current population to produce the next generation according to the results of the *evaluation* step; the idea is

to consider the ones with the best evaluation results. In this case, we select the 50 % of the chromosomes with the best fitness score.

4. *Reproduction*: in this step, two individuals are recombined through genetic operators like crossover and mutation to create a new individual; the chromosomes are randomly pairing; each pair produces two offspring that contain traits of each parent.

We intend to use this method as an alternative to discover a possible combination of weights that work better during the distribution process. Although assigning equal value to each canon of justice seems to be the best choice, it is necessary to verify it with experimental analysis.

4.6. Putting all the pieces together

In order to generate a computational executable specification that combines the principles for enduring institutions, the cooperation models based on stochastic games, the Rescher's canons of distributive justice and the adaptative mechanisms presented in Section 4.5, we use the framework for institutional analysis introduced by Ostrom as the internal architecture of a multi-agent system [143]. This framework enables us to use the notion of action situations to create a relationship among the set of agents, the physical attributes of the environment, and the institutional agreements related to the distribution process. The proposed architecture is presented in Figure 4-2. In this model, an action situation represents any point in time and space in which a set of agents can cooperate for managing and sharing common-pool resources. Accordingly, we can define a multi-agent system as follow:

$$M = \langle N, A, I, \xi \rangle \quad (4-10)$$

where:

- N is the set agents;
- A is the set of action situations;
- I is the set of institutions; and
- ξ represents the physical attributes of the environment.

Each action situation has a number of roles according to the institutional rules; in this case, we use the roles of *prosumer* and *head*: the *prosumer* determines which agents have access to institutional rights. The *head* is responsible for performing the allocation of resources. Furthermore, each action situation $a_i \in A$ can be defined by:

$$a_i = \langle c, \ell, g \rangle \quad (4-11)$$

in which:

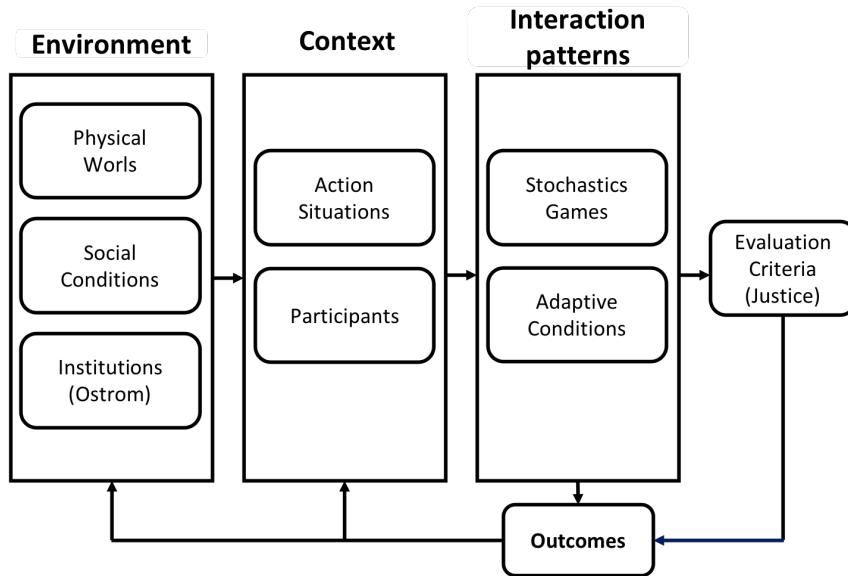


Figure 4-2.: An institutional approach for resources distribution

- c represent the set of agents related to the action situation a_i , such that $c \subseteq N$;
- ℓ is a specification of I ; and
- g is a game that describes cooperation pattern in the system.

The institution specification ℓ is defined through two levels of nested rules based on the Ostrom's approach. The first level, the *operational-choice rules*, is represented as a function that maps a set of demands to a set of allocations. This function is exchangeable and allows us to explore different distribution methods like legitimate claims, equality, random allocation, maximin and so on. The second level, the *collective-choice rules*, combines the agents' preferences and compute new weights of each canon justice during the allocation process. The voting system is also exchangeable and independent of the institution's structure. The combination of all these elements enables us to do a meta-analysis of the distribution problem not only in terms of the main components involved in the process but also regarding the information flows in the system. The model includes a set of exchangeable parameters based on the idea of action situation that allows us to maintain the structure of the multi-agent system no matter if the cooperation pattern, the allocation method, the adaptive mechanisms, or the voting system are changed.

4.7. Experimental results and evaluation

In this Section, we evaluate the effectiveness of the proposed model using a set of experiments to analyze different aspects related to the distribution process. For each scenario,

several allocation methods, voting systems, and adaptive mechanisms are used in order to assess the performance of the model and verify if the system behaves as expected. We structured this part as follows: Section 4.7.1 presents the configuration parameters common to all experiments; Section 4.7.2 describe the purpose of each simulation scenario and Section 4.7.3 reports the obtained results. The full source code is available for download¹.

4.7.1. Experimental method

To implement and test the simulations, we use the simulator NetLogo, which is a multi-agent programmable modeling environment widely used in teaching and research of complex systems. It is essential to mention that not only our method is tested, but also previous works for comparison purposes. The following configuration parameters are used unless stated otherwise:

- The simulations are composed of two different types agents: on the one hand, compliant agents who are willing to follow the institutional rules, appropriating what is allocated by the institution. On the other hand, selfish agents who can cheat on the demand, provision, or appropriation actions according to a probability value of 0,25; this probability represents malicious repeated behaviors in the system.
- The Prisoner’s dilemma uses the following parameters value: $b_1 = 2$, $c = 1$ and $k = 0,4$.
- We use a population of 30 agents in the experiments; this number allows us to observe global patterns of behavior and the possible effects of the selfish agents in the system.
- An agent i is formed of a set of variables that define its behavior; in this case, q_i, d_i, p_i, g_i, r_i and r'_i represent need, demand, provision, availability, allocation, and appropriation of resources in the current round. These variables depend on the cooperation pattern and are used according to the games presented in Sections 4.3.1 and 4.3.2.
- S represents the agent’s satisfaction and u the agent’s utility; both are updated after each round.
- $\alpha = 0,1, \beta = 0,1, \tau = 0,1, m = 3, S(0) = 0,5$ are stated as the initial values for all agents in the simulations. These parameters determine if an agent keeps following the institutional rules; if for m consecutive rounds, the agent i evaluate $S_i < \tau$ as true, it will stop cooperate. These values represent a moderate behavior in which an agent will wait for several rounds without satisfactory results before stopping cooperate. Values of $\beta = 0,4$ are used to represent agents with more volatile behavior. These parameters are selected according to the results presented in [1].
- The agents’ behavior is defined as follows: compliant agents will demand what they need ($d = q$), provision what is available ($g = p$), and appropriate what is allocated

¹<https://github.com/jospinalo/ComputationalJustice>

$(r' = r)$. On the other hand, selfish agents could increase its demand above what it needs $d = q + \text{rand}(0, 1) * (1 - q)$; reduce their provision of resources $p = g * \text{rand}(0, 1)$; or increase the quantity appropriated above what has been allocated by the institution $r' = r + \text{rand}(0, 1) * (1 - r)$.

- A resource is modeled by a variable $R \subseteq \xi$, which represents the available resources in the environment or the sum of the agents' provision according to the cooperation pattern. Furthermore, to fulfill the second design principle for enduring institutions, we ensure that the expected value of the sum of all agent's needs is equal to the available resources in the system. This condition not only maintains the scarcity condition but also ensures that in the long-term, there are enough resources to keep satisfied all the agents in the system. It is important to mention that without this principle, this problem turns into a viability problem instead of a distribution problem.
- Each experiment was repeated 50 times using different random seeds to keep randomness in the simulation.

To analyze the data related to the experiments, we use the following metrics:

- *Average utility*: \bar{u}_c and \bar{u}_s represent the average utility of the agents during the simulations; \bar{u}_c correspond to the compliant group and \bar{u}_s to the selfish group.
- *Fairness*: we evaluate the fairness of distribution process using the Gini index over $(\sum_{i=0}^T r_i / \sum_{i=0}^T d_i)$; an index of 0 represents perfect equality, and 100 represents perfect inequality. This metric is usually used as a statistical measure of the distribution of resources in social systems.
- *Average satisfaction*: \bar{s}_c and \bar{s}_s represent the average satisfaction of the agents during the simulations; \bar{s}_c correspond to the compliant group and \bar{s}_s to the selfish group.

4.7.2. Simulation scenarios

We performed several sets of experiments to evaluate the behavior of the distribution method; in each case, we use both cooperation patterns mentioned in Section 4.3.

– *Scenario 1: evaluating different allocation strategies*. This scenario has the purpose of verifying if our model achieves better results than other distribution schemes. We consider the following methods for comparison purposes:

- A. *Random-appropriation*: each agent appropriate a random proportion of the available resource; this process is repeated until the resource is depleted.
- B. *Random-allocation*: the institution performs a random allocation among the agents without considering their demands; the sum of all allocations is equal to the available resources in the system.

- C. *Equal-allocation*: the institution allocates an equal proportion of resources to each agent.
- D. *Computational-justice*: the institution performs the allocation according to the Researcher's approach for distributive justice. In this case, we use an equal and fixed weight for each canon justice.

—*Scenario 2: adaptive strategies*. In this scenario, we evaluate how selfish agents affect the performance of the distribution method regarding cheating behaviors in demand, provision, and appropriation actions. Also, we aim to verify the effect of the adaptive strategies mentioned in Section 4.5. We use these methods to modify the weight of each canon of justice and allow the institution to adjust their parameters to possible changes in the agents' behavior.

—*Scenario 3: volatile agents*. In this scenario, we explore the performance of the model in simulations composed by agents with a higher volatile behavior, i.e., agents who decrease faster their satisfaction level if the institution does not fill their demands. Also, we use adaptive strategies to verify if they represent an opportunity to deal with these behaviors.

4.7.3. Experimental results and performance evaluation

The results presented in this Section were generated by running the simulation scenarios described above. After each experiment, we analyze the obtained data and compare them with expected results.

Scenario 1: evaluating different allocation strategies

In this set of experiments, we compared the evaluation of different distributions schemes; all scenarios were composed of complying agents. Table 4-3 and Figure 4-3 show the results for the cooperation pattern based on the prisoner's dilemma; Table 4-4 and Figure 4-4 show the results for the cooperation pattern based on the linear public good game. In both cases, the results show how an equal distribution is inappropriate under a scarcity condition. This affirmation coincides with the analysis presented in [1], and controvert the idea in which allocating the same amount of resources is a good alternative to face the distribution problem in self-organizing open systems. It is essential to remark that the agents' satisfaction depends on the capacity of the institution to satisfy the agents' needs, and not just for a particular allocation method that seems to be fair from the system level. Additionally, even though the Gini-index shows a proper fairness evaluation for the equal allocation, the agents' utility is affected because of the institution does not consider their demands; this situation affects the endurance of the cooperation process in the long-term.

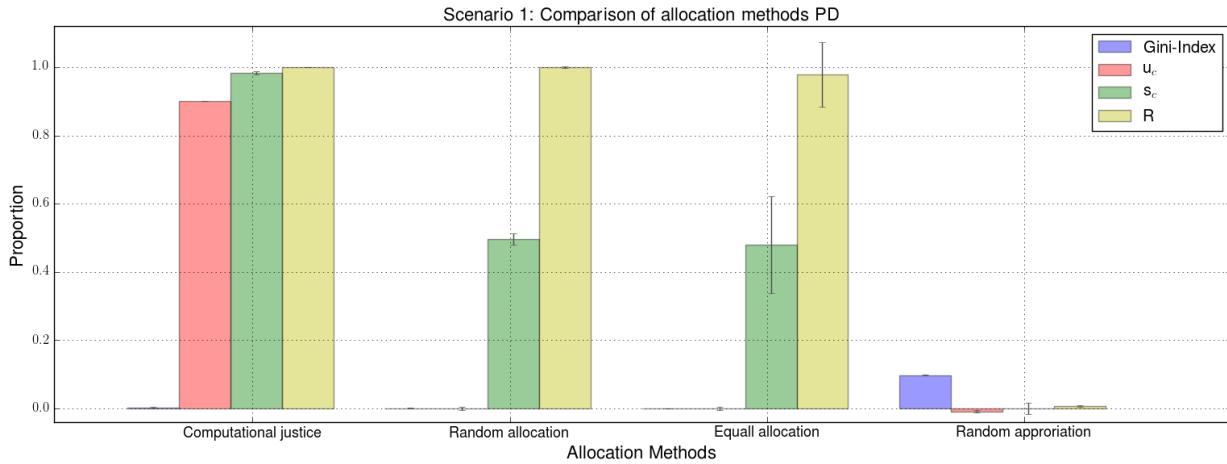


Figure 4-3.: Scenario 1 PD: comparison of allocation methods

Table 4-3.: Scenario 1 PD: comparison of allocation methods

Allocation Method	Gini-index	u_c	s_c	Round	R
Computational justice	0,0029 ± 0,0005	0,9002 ± 0,0006	0,9835 ± 0,0040	2500	1,00 ± 0,00
Random allocation	0,0013 ± 0,0002	0,0002 ± 0,0011	0,4960 ± 0,0044	2500	0,99 ± 0,01
Equal allocation	0,0000 ± 0,0000	0,0008 ± 0,0014	0,4798 ± 0,0393	2500	0,98 ± 0,03
Random appropriation	0,0980 ± 0,0002	-0,0084 ± 0,0011	0,000 ± 0,044	2500	0,01 ± 0,01

Similarly, the notion of sustainability can be defined according to the cooperation pattern. In the context of the prisoner's dilemma, we represent this attribute as the availability of resources at the end of the game. Although both the random and the equal allocation methods result in a sustainable distribution process, the computational justice model shows a better performance regarding the satisfaction of the agents. In the context of the linear public good game, sustainability is represented as the number of players that remain at the end of the simulation. In both scenarios, the computational justice model achieved better performance in comparison with the other allocation methods. Table 4-4 shows the final round for each scenario.

On the other hand, the random appropriation method showed the worst performance in terms of the agents' utility and the sustainability of the cooperation process. These results show how the lack of a suitable allocation method can lead the system to a social dilemma and also exhibit the high sensibility of this proposal to appropriation cheating actions. However, these results show how an allocation method based on computational justice is a potential alternative to deal with the distribution problem in self-organizing open systems.

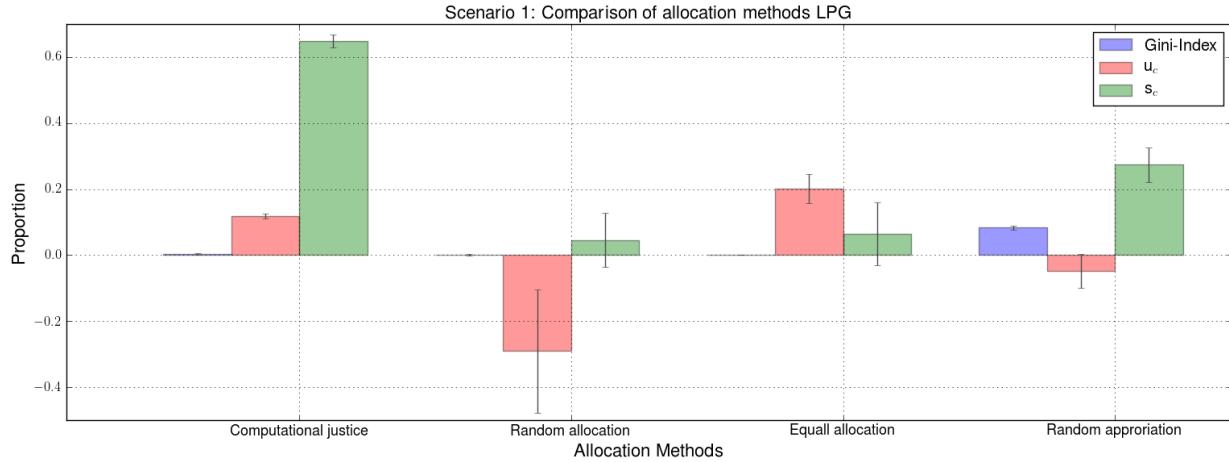


Figure 4-4.: Scenario 1 LPG: comparison of allocation methods

Table 4-4.: Scenario 1 LPG: comparison of allocation methods

Allocation Method	Gini-index	u_c	s_c	Round	N_c
Computational justice	0,0040 ± 0,0029	0,1177 ± 0,0077	0,6488 ± 0,0192	2500	30
Random allocation	0,0006 ± 0,0007	-0,289 ± -1,871	0,0461 ± 0,0225	19	0
Equal allocation	0,0000 ± 0,0000	0,2012 ± 0,0121	0,653 ± 0,0264	97	2
Random appropriation	0,084 ± 0,0018	-0,0474 ± 0,0143	0,2742 ± 0,0144	527	2

Scenario 2 PD: selfish agents

To see how our proposal reacts to selfish agents, we analyze the behavior of the model through the three possible cheating strategies: demanding more than is needed, appropriating more than is allocated, and provisioning less than is available. The experiments were performed with a population of 30 agents (21 compliant and 9 selfish), playing a game for 2500 rounds. It is the typical population used to analyze scenarios in which selfish agents can have a significant impact on the performance of the model [1]. Moreover, we include both adaptive strategies as part of the institution's behavior to adjust the weight of each canon of justice and verify if these methods result in better control of cheating behaviors. Table 4-5 presents the results for the cooperation pattern based on the prisoner's dilemma; Table 4-6 shows the results for the cooperation pattern based on the linear public good game.

In both cases, the experiments show how computational justice can deal with cheating behaviors related to provision and demand. In the context of the prisoner's dilemma, sustainability is not affected keeping a significant part of the resources at the end of the game. On the other hand, in the linear public good game, the computational justice model promotes cooperation prioritizing the allocation over compliant agents and gradually forcing selfish agents to leave

Table 4-5.: Scenario 2 PD: adaptive strategies

Strategy	Cheating Method	Gini-index	u_c	u_s	s_c	s_s	R
Fixed	Demand	$0,01 \pm 0,02$	$0,71 \pm 0,03$	$0,27 \pm 0,03$	$0,66 \pm 0,12$	$0,01 \pm 0,02$	$0,83 \pm 0,07$
	Appropriation	$0,10 \pm 0,02$	$-0,06 \pm 0,01$	$-0,06 \pm 0,01$	$0,0 \pm 0,0$	$0,0 \pm 0,0$	$0,03 \pm 0,01$
Voting	Demand	$0,01 \pm 0,01$	$0,58 \pm 0,07$	$0,22 \pm 0,03$	$0,34 \pm 0,13$	$0,26 \pm 0,09$	$0,70 \pm 0,08$
	Appropriation	$0,07 \pm 0,02$	$-0,06 \pm 0,01$	$0,06 \pm 0,01$	$0,0 \pm 0,0$	$0,0 \pm 0,0$	$0,01 \pm 0,01$
GA	Demand	$0,01 \pm 0,02$	$0,73 \pm 0,07$	$0,29 \pm 0,03$	$0,72 \pm 0,11$	$0,52 \pm 0,08$	$0,85 \pm 0,17$
	Appropriation	$0,07 \pm 0,02$	$-0,06 \pm 0,01$	$0,06 \pm 0,01$	$0,01 \pm 0,01$	$0,0 \pm 0,0$	$0,03 \pm 0,01$

Table 4-6.: Scenario 2 LPG: adaptive strategies

Strategy	Cheating Method	Gini-index	u_c	u_s	s_c	s_s	N_c	N_s
Fixed	Demand	$0,01 \pm 0,01$	$0,11 \pm 0,01$	$-0,06 \pm 0,02$	$0,64 \pm 0,01$	$0,43 \pm 0,07$	21	2
	Appropriation	$0,01 \pm 0,01$	$0,14 \pm 0,01$	$-0,01 \pm 0,01$	$0,64 \pm 0,01$	$0,6 \pm 0,1$	18	6
	Provision	$0,10 \pm 0,01$	$0,08 \pm 0,04$	$0,08 \pm 0,02$	$0,64 \pm 0,01$	$0,52 \pm 0,05$	21	2
Voting	Demand	$0,01 \pm 0,02$	$0,11 \pm 0,01$	$-0,05 \pm 0,02$	$0,63 \pm 0,01$	$0,40 \pm 0,08$	19	1
	Appropriation	$0,01 \pm 0,02$	$0,12 \pm 0,01$	$0,02 \pm 0,01$	$0,64 \pm 0,01$	$0,61 \pm 0,01$	18	6
	Provision	$0,01 \pm 0,01$	$0,08 \pm 0,01$	$0,05 \pm 0,02$	$0,63 \pm 0,01$	$0,26 \pm 0,08$	19	1
GA	Demand	$0,01 \pm 0,01$	$0,12 \pm 0,01$	$-0,08 \pm 0,02$	$0,64 \pm 0,01$	$0,50 \pm 0,01$	21	2
	Appropriation	$0,01 \pm 0,02$	$0,13 \pm 0,01$	$0,01 \pm 0,01$	$0,65 \pm 0,01$	$0,61 \pm 0,01$	19	1
	Provision	$0,01 \pm 0,01$	$0,08 \pm 0,01$	$0,07 \pm 0,01$	$0,64 \pm 0,01$	$0,51 \pm 0,06$	21	2

the system. In contrast, scenarios composed of cheating appropriators show the ineffectiveness of our proposal to deal with this behavior; in both cases, the cooperation process does not endure and lead the system to a social dilemma. These results can be understood as a consequence of the scarcity condition and show the need to develop alternative mechanisms to face this situation. A possible solution could be using retributive justice to control and punish misbehave agents as is suggested in [60].

Furthermore, the adaptation strategies do not represent a better performance in comparison with a fixed weights scenario. Our proposal exhibit robustness regarding cheating actions in provision and demand and allow the institution to perform an allocation without the additional computational cost that represents to include adaptive mechanisms in the model. This result denotes an improvement in comparison with the initial proposal for computational justice [1], in which an institution with fixed weights is not able to deal effectively with these types of behaviors. It is important to mention that this improvement appears because of a variation in the equality canon of justice, as we explained in Section 4.4.

Scenario 3: volatile agents

In these set of experiments, we analyze the performance of the model in environments composed by volatile agents. The aim is to verify how the distribution model responds to situations in which the satisfaction level of the agents decrease faster if the institution does not fulfill

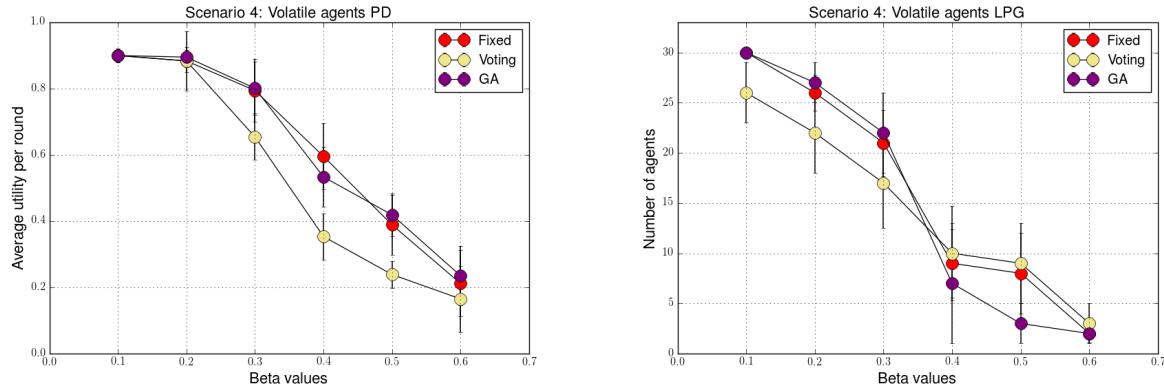


Figure 4-5.: Scenario 3: volatile agents for PD and LPG

their demands. Besides, we use the adaptive strategies presented in Section 4.5 to confirm if these methods are useful in these scenarios. We selected values of $\beta \in [0,1,0,6]$ to include agents with a lower tolerance to not getting what they need. Figure 4-5 present the result of both adaptive strategies in comparison with a fixed weights scenario.

In both cases, the results show how the utility of the agents decreases with high values of β , and depict how the distribution process has a certain level of dependency on stable social conditions. In an environment in which the agents turn into free-riders faster, the institution does not have the opportunity to stabilize the distribution process creating a dissatisfaction reinforcement that affects the performance of the system. Moreover, although the use of adaptive strategies describes a performance improvement for some values of β , these methods do not represent a significant enhancement if we consider the additional computational cost associated with introducing these methods as part of the institutional structure. As a result, we need to consider the behavior of the agents as an essential input for the distribution process, no matter the allocation rules used by the institution.

4.8. Conclusions

In this chapter, we proved that it is possible to deal with the distribution problem in the self-organizing open system using a combination of socially inspired computing and agent-based modeling. In particular, this proposal was composed of: the Ostrom's approach for institutional analysis, cooperation patterns through stochastic games, the Rescher's idea of distributive justice, and adaptive mechanisms through voting systems and genetic algorithms. The result showed how an allocation process based on computational justice is a potential alternative to deal with the distribution problem in environments formed by autonomous agents operating without a central controller or other orchestration forms. It is essential to mention that this

work was developed on the foundation of self-organizing electronic institutions established by Pitt et al. [1], in which they showed the potential value of the distributive justice and social institutions as part of the future technological developments.

On the other hand, the notion of action situation allows us to maintain the structure of the multi-agent system no matter if the cooperation pattern, the allocation method, the adaptive mechanisms, or the voting system are changed. This structure enables us to do a meta-analysis of the distribution problem not only in terms of the main components involved in the model but also regarding the information flows in the system. Moreover, the inclusion of stochastic games as part of the cooperation patterns results in a suitable choice for modeling the relationship between the environment state and the agents' actions; this feature gives us a more realistic version of the distribution problem and results in better cooperation levels. Furthermore, we modified the canon of equality introduced in the initial proposal of computational justice, producing a distribution method that can deal with cheating actions in provision and demand without the use of adaptive strategies; this feature decreases the computational cost involved in the allocation process and represents an improvement regarding previous works.

In conclusion, we have shown how it is possible to create computational models using social and biological systems as an inspiration source. In particular, the distribution problem is perhaps as ancient as human society and draws an opportunity to use social concepts as an input for developing new algorithms for self-organizing communication networks. Furthermore, some parameters in this proposal need to be considered for future research; for example, selfish agents with multiple strategies, new values for the cooperation patterns, variations of the current adaptive methods, and different approaches for distributive justice. We expect the inclusion of these elements allows us to model a self-organized communication network as a social system.

5. A self-organizing multi-agent system for ad hoc networks

5.1. Introduction

A big challenge during the process of implementing ad hoc networks is to endow them with autonomous, scalable, and adaptative behaviors. These conditions create the need to design and implement computational mechanisms to allow the network to adapt its parameters to possible changes in the operating conditions. Similarly, all artificial systems use management and control processes to control their behaviors. The management process consists of manipulating subsystems, parameter updates, and verify the system state. Control is about feedback and run-time control based on the possible variations in the environment. Both processes are required to maintain, operate, and adapt the system to unexpected situations. In this regard, the agent paradigm provides a conceptual framework to face the challenge of implementing ad hoc networks using self-organization as a design principle. In this chapter, we propose a multi-agent system as a computational framework for developing ad hoc network applications based on socially and biologically inspired computing. We aim to provide an easy and low-cost mechanism for implementing ad hoc networks through the agent paradigm.

The rest of this chapter is organized as follows: Section 5.2 reviews the basic ideas and the most common applications of ad hoc networks. Section 5.3 presents the two different approaches in the agent paradigm: the agent-based modeling for analyzing complex phenomena and the multi-agent system for implementing engineering artifacts. In Section 5.4 the proposed architecture for the multi-agent system is presented according to the *Structured analysis and design technique* (SADT) methodology. Section 5.4.1 describes the artificial agents using structural and functional perspectives. In Section 5.5, we explored applications like coverage expansion, sensor networks, and mobile clouds. Finally, Section 5.6 concludes the chapter.

5.2. Ad hoc networks: a brief overview

Ad hoc networks are self-organizing systems composed of autonomous nodes of different provenance operating without a central controller [14]. Nodes, users, and services can compete

or cooperate to achieve individual and collective goals. Thus, it is reasonable to assume a common language, and a mutually agreed set of rules that specify the acceptable behaviors in the system [31]. It is possible to identify six critical features in ad hoc network [60, 39]:

- *Cooperation and competition*: nodes, users, and services depend on cooperation processes to achieve their goals. However, network components could be unreliable and may have conflicting aims. For example, if they are competing for the same resources;
- *Mutability*: the environment, the topology, and the system state can change frequently and unpredictably; ad hoc networks have highly dynamic operating conditions, and also each component has to deal with first encounter problems;
- *Partial and uncertain knowledge*: because of interactions are asynchronous, in parallel, or peer-to-peer, and no one has complete and reliable information about the system state. Each component only has a partial (and possible subjective) knowledge of the overall system. Besides, the union of this knowledge may be inconsistent;
- *Self-organization*: There is no central controller or other orchestration forms; nodes have to resolve differences, deal with first encounters, cope with uncertainty, and recover from errors by and between themselves;
- *Endogenous resources*: The nodes provide all the resources according to their computational capabilities. There are no external resources that can be used by the system, so it is reasonable to assume an economy of scarcity;
- *Error expectation*: Due to competition, conventional rules, and high levels of autonomy, behavior contrary to the specification should be expected. Errors may result from accidents, need, or design (free-riders). So, it is necessary to distinguish between their causes and how to recover from them.

Two reasons can lead to the use of an ad hoc network: the unavailability of infrastructure or high costs related to its deployment. There are scenarios in which it is impossible to have a network infrastructure [18, 14]. For example, emergency situations like floods or hurricanes in which the rescue process requires a quick setup of a communication network. Similarly, ad hoc networks are useful when it is possible to avoid unnecessary costs for deploying a traditional network infrastructure. For instance, in rural sites or in construction areas in which it is too expensive or inconvenient to set up a network just for temporary use. Another scenario could be collaborative and distributed computing for quick data transfers and resource sharing among collaborating people, e.g., during a conference, in a classroom, or for deploying a mobile cloud [61]. Some applications of the ad hoc network are presented in Figure 5-1.

First, VANETs are formed spontaneously and have short lifetimes [62, 39]. Cars work as nodes connected through a wireless channel with the purpose of exchanging information about traffic, roads, or weather. VANETs have variations in the number of nodes and highly

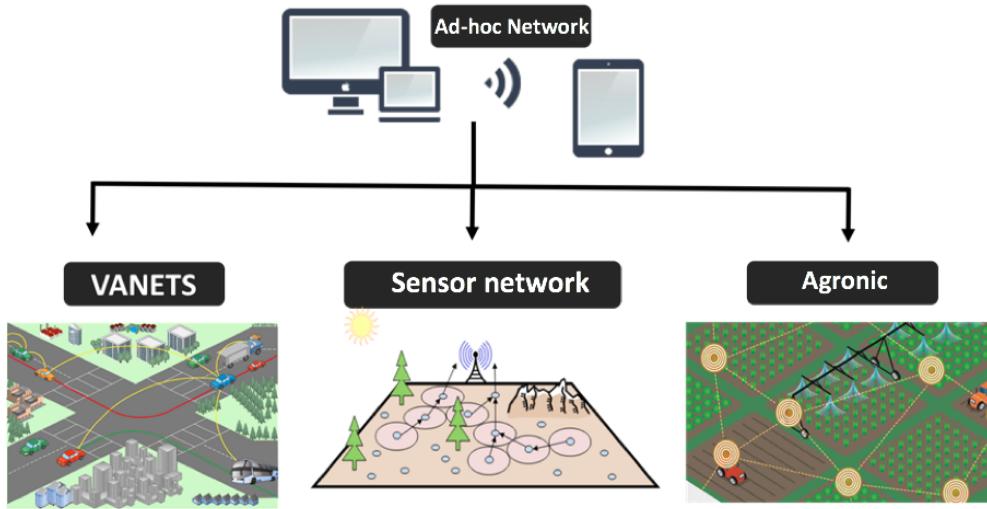


Figure 5-1.: Ad hoc networks applications

dynamic topologies. As a result, they face challenges related to routing, data dissemination, and coordination for vehicle platooning. Secondly, WSNs are used for collecting and processing data in environments without infrastructure [48, 57]. Typical applications of WSNs are environmental monitoring, emergency networks, and smart cities. All these scenarios include multi-hop topologies, resource constraints, and different mobility patterns. Finally, there is a growing research area related to precision agriculture in which communication networks are used to observing, measuring, and responding as fast as possible to environmental conditions in crops [150, 151]. The goal of ad hoc networks in this scenario is to provide a decision-making support system to optimize the production process in an environment without a pre-established communication infrastructure. It is important to mention that regardless of the application, all cases share similar conditions like dynamic nature, lack of centralized control, autonomous operation, and a need for solving problems through collective actions. Thus, we can use biologically and socially inspired computing for modeling and implementing ad hoc networks enough robust and adaptive to operate in the scenarios mentioned above [59].

5.3. The agent paradigm

The use of the agent paradigm for studying complex systems is a significant and growing research area in natural science and engineering [152]. There are two agent approaches: agent-based modeling (ABM) and multi-agent systems (MAS) both with different methodologies, application, and goals [153, 154]. In the first place, ABM is used for representing and simulating the actions of a group of autonomous agents, and capture their dynamic for

analytical purpose. It combines artificial intelligence, game theory, computational sociology, and concepts like emergence and self-organization to find explanatory models for collective behavior. In the second place, MAS focus on solving practical engineering problems through artificial agents acting and affecting a physical environment. MAS uses software agents for managing and controlling computational infrastructure related to communications networks, smart grids, and transportation systems. Nevertheless, a multi-agent system could be composed by a mixing of robots, computational infrastructure, software agents, and humans.

Similarly, to translate biological and social principles into ad hoc networks, ABM and MAS can be used as part of the methodology presented in chapter 2. First, ABM allows to capture and understand the essential dynamic of different biological and social phenomenon (representation and modeling stages). Second, MAS allows us to develop software agents (application stage) for networking applications in which the agents exhibit the behaviors modeled through ABM. It is essential to mention that social behaviors are fundamental in both ABM and MAS. In ABM, the agents usually represent individuals and their possible social interactions. In MAS, sociability is required for solving problems through collective actions like coordination, cooperation, and collaboration. In this research, ABM was used to analyze the distribution problem and propose an allocation method based on computational justice. MAS is used for implementing our proposal in the context of ad hoc networks.

5.4. A self-organizing multi-agent system for ad hoc networks

It is possible to use the agent paradigm for modeling an ad hoc network as an interaction network of two levels. First, the physical level represents the network devices and their wireless connection operating in a self-organizing manner. Second, the software agents and their relationships create an interaction network at the logic level. The agents can be grouped based on their goals and behavior to create coalitions related to specific network functionalities. In this context, biologically and socially inspired computing take a fundamental role because they provide computational mechanisms to promote autonomy, sociability, and adaptative behaviors in the system. It is expected by the agents and coalitions to exhibit the behaviors described in chapter 2 for controlling and managing the system.

Formally, a multi-agent system can be defined as a fixed and finite set $N = \{1, 2, 3, \dots, n\}$ of agents inside an environment. A coalition, C , is simply a subset of N . The grand coalition is the set of all agents N . The idea of coalition implies some common purpose or commitment to collective goals. Figure 5-2 describes the expected behavior of a multi-agent system. We

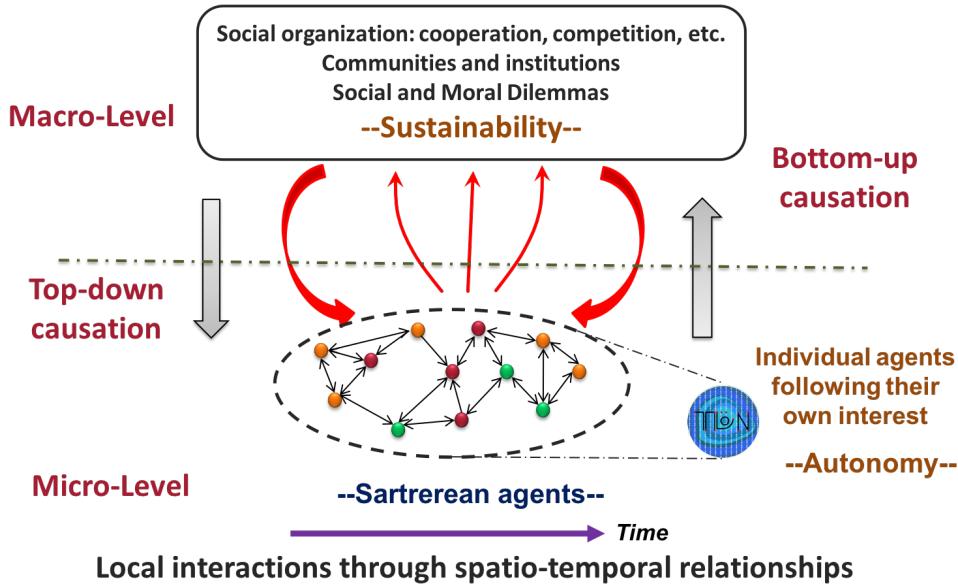


Figure 5-2.: Internal dynamics of a multi-agent system

need to consider two levels of analysis. In the first place, the system level (or a macro-level) wherein all agents and coalitions are combined, and they cooperate or compete for achieving an overall outcome. At this level, we find properties like sustainability, system functionalities, fairness, emergent properties, or any measure that allows analyzing the system as a single unit. In the second place, we have the agents level (or micro-level) wherein autonomous agents are pursuing their agendas according to a self-cost-benefit analysis. The agents' strategies, goals, actions, and individual behaviors are found at this level. As a result of the agents' goals, a value system is created for describing any resource that the agents can use to improve their status according to the system state. For example, in the context of self-organizing networks, these resources can be represented as information, CPU, memory, energy, or collaboration. Additionally, for designing a MAS, we need to define the following elements: the agent architecture, their social organization, the interaction patterns, and the environment. A detail description of these elements is presented below.

5.4.1. Artificial agents: functional and structural perspectives

The first step for building a multi-agent system is to define how we are going to build at least one of them. In this regard, it is necessary to specify how an agent can be modeled as a set of software components and explain what is expected in terms of behavior and structure. An agent can be defined as a computer system that is situated in an environment and is capable of autonomous action. Although there are different computational approaches for designing and implementing agents, we can establish some of their expected features:

- *Autonomy*: This is an essential attribute for the notion of agency. It can be defined as the quality or state of being self-governed. However, there is a spectrum in the notion of autonomy that we need to consider. At one extreme, there are human beings; we have as much freedom as anybody does concerning our beliefs, goals, and actions. On the other hand, we have simple software applications; they do what they were programmed to do. However, there are many possibilities between these two extremes. We explore these possibilities in Section 5.4.1 based on the idea of cognitive states.
- *Reactivity*: An agent can perceive signals of its environment and respond appropriately to changes that occur in it in order to satisfy its designing goals. The central role of reactivity is to define an interaction pattern that allows us to describe the agents' behavior according to a set of states and the possible relationships among them.
- *Proactivity*: It is expected for artificial agents to exhibit goal-directed behavior by taking advantage of the opportunities available either in the environments and in their social group. The intention is design systems that not only respond to environmental changes but also take actions autonomously for achieving their goals without the need of an external stimulus.
- *Social ability*: Artificial agents are capable of interacting with other agents in order to satisfy their designing goals. Social abilities are needed in environments formed by a set of co-dependent components in which there is a tension between individual and collective rationality. In this case, the agents need to compete, cooperate, and negotiate with others for achieving individual and collective goals.
- *Adaptation*: We can define adaptation as the process in which a system changes to cope with variations in the environment. These changes can be a result of any alteration in the structure or function of the system. Adaptive behaviors are useful to deal with non-stationary problems where the searching space is changing continuously.

A structural perspective: Sartrean model

Figure 5-3 provides a model to represent the structure of a living organism from two different points of view: Biological and Social. On the one hand, we can analyze a living organism from the biological perspective through three different levels of organization: *Phylogeny* (P), *Ontogeny* (O), and *Epigenesis* (E) [115, 113]. First, *Phylogeny* is related to the temporal evolution of the genetic program. This process is fundamentally non-deterministic and gives rise to the emergence of new organisms through recombination and mutation of the genetic code. Second, *Ontogeny* is related to the development of a single individual from its genetic material. Finally, *Epigenesis* is concerned about the learning process in which an organism can integrate information from the outside world through interactions with the environment. We can understand the POE model as follows: *Phylogeny* involves evolution; *Ontogeny* involves development, and *Epigenesis* involves learning. In this regard, evolutionary computing can

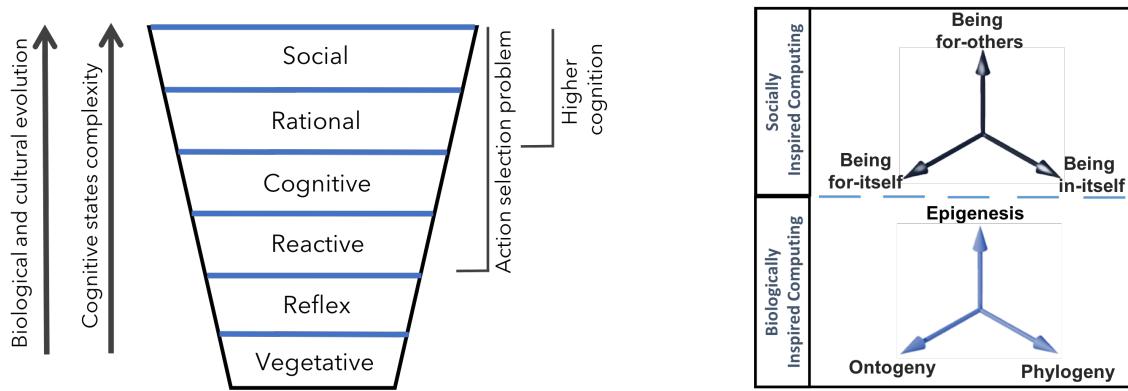


Figure 5-3.: Agents' architecture

be seen as a simplified artificial counterpart of Phylogeny in nature. Multicellular automata, self-replicating, and self-healing software are based on ontogeny properties. Finally, artificial neural network and artificial immune systems can be seen as examples of epigenetic processes.

On the other hand, at the social level, we can extend the biological perspective using a social approach. We can represent this level through three new axes. The idea is to provide an expansion of the POE model. The axis are: *being for-itself*, *being in-itself* and *being for-others*. *Being for-itself* allows us to encapsulate the desires of an agent; it refers to its capacity to define a goal and or a purpose by itself. *Being for-other* allows us to encapsulate complex social patterns like cooperation, competition, altruism, and collaboration. Finally, *being in-itself* is related to being conscious; a set of behaviors that allow an agent to reason about its goals, actions, needs, and internal states. It is important to mention that these three levels assume higher levels of cognition. We can summarize as follows: *Being for-itself* as cognition, *being in-itself* as metacognition and *Being for-others* as sociability. This model, known as *Sartrean agent*, was proposed by Ortiz J. and is the basis of the Tlön Project¹.

A functional perspective: cognitive states

Figure 5-3 offers a classification of the behaviors exhibited by living organisms based on the complexity of their cognitive state. Accordingly, we can classify the behaviors of an agent in the following types: *vegetative*, *reflex*, *reactive*, *motivated*, *rational*, and *social*. The *vegetative* behaviors are the ones that are in an organism by default; for example, breathing, metabolizing, and so on. In the context of artificial agents, these behaviors are related to the implicit internal functions of a software application. Usually, these behaviors are not considered by an external observer because they are always there. *Reflex* behaviors are stimulus-response-based. If there is a stimulus in the environment, the agent always produces an action as a

¹A detail description can be found at <http://www.tlon.unal.edu.co/proyecto-tlon/modelo-social>

response. *Reactive* behaviors are also stimulus-response-based. However, reactive behavior includes an action selection process, which may cause the action will not be executed despite the presence of the stimulus. Finite state machines are usually used to represent this kind of behavior

Motivated behaviors are those that need both an external stimulus and internal motivation. We assume as internal motivation the stimulus related to the individual goals and needs of the agent. *Rational* behaviors are usually linked to decision models and the use of symbolic language for representing and manipulating information through logical axioms during the action selection process. Furthermore, in situations in which the agent can reasoning about what he is reasoning, i.e., it is aware of its behaviors and actions, we can think about him as a conscious agent. Both rational and conscious behaviors require higher levels of cognition. It is important to mention that the boundaries between different types of behaviors are fuzzy, and we cannot define them accurately.

Traditionally, the notion of rationality has been used in artificial intelligence to model decision-making processes and reasoning schemes. They aim to produce computational mechanisms that allow the evaluation and execution of actions in order to maximize the goals of an individual. We can analyze rationality from two different scenarios. First, situations in which a single agent is acting in an environment that could be unknown, stochastic, partially observable, and so on. It could be difficult to find an optimal strategy, but there is a well-defined notion of what an optimal strategy is. In this case, decision theory provides a set of formal techniques to analyze individual decision process. Second, situations composed by multiple agents in which everyone is trying to adapt their strategies and achieve their goals at the same time. When an agent adapts its behavior, it is influenced not only by the environment but also by the behavior of other agents. We can use game theory for modeling collective behaviors based on cooperation and competition.

Finally, we have social behaviors; in particular, we are interested in them because they are the basis of collective actions. Cooperation, competition, and negotiation depend on the ability of an agent to recognize other agents in the environment and also be able to interact with them as part of a social group. Although both living organisms and societies can be considered as meta-systems, the difference between them is the level of autonomy in their components. While the units of an organism have little or no independence, those in a social system have a maximum level (for instance, human society). Also, we can recognize complex social patterns like cooperation, institutions, symbolic language, and justice, in environments in which agents develop highs level of sociability. Our goal is to produce these kinds of behaviors as emergent properties in the system. Since we want to use sociability as a way to increase

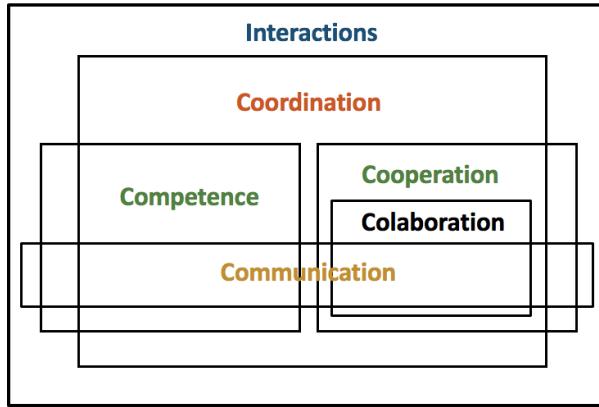


Figure 5-4.: Interaction patterns in multi-agent systems

the capacity of the system to solve problems to collective actions, it is needed to define the possible social behaviors of the agents. Figure 5-4 provides a classification of them based on the notion of interaction.

5.4.2. Multi-agent system: proposed architecture

For describing the structure of the multi-agent system, we use the *Structured analysis and design technique* (SADT), which allows us to make a top-down decomposition. The idea is using blocks to represent entities and activities to perform a functional examination of the system. Figure 5-5 presents a perspective of the second level of analysis. Appendix A provides a complete description of the proposed architecture, and also the full source code is available for download². Accordingly, the multi-agent system is formed by seven components: *Natural laws*, *Cultural laws*, *Register*, *Communication*, *Resources*, *Agent factory* and *World*. The *Natural laws* refers to any configuration or parameter associated with the environment. It defines the shape of the system, i.e, the rules which determine how the different components will be related. It is important to mention that in the current stage of this research, we are interested in *Cultural laws*, leaving a detailed design of *Natural laws* for future work.

Similarly, the *Agent factory* allows us to implement software agents according to the model presented in Section 5.4.1. Reflex and reactive behaviors use finite state machines to describe different states and their possible relationships. Rational behaviors use decision theory and optimization techniques as part of their action selection process. Social behaviors are modeled using game theory through cooperative, non-cooperative, and coalitional games. They aim to create scenarios based on cooperation and competition to solve problems through collective actions. Each agent is programmed as a process which has embodied one or more

²https://github.com/jospinalo/MAS_TLON

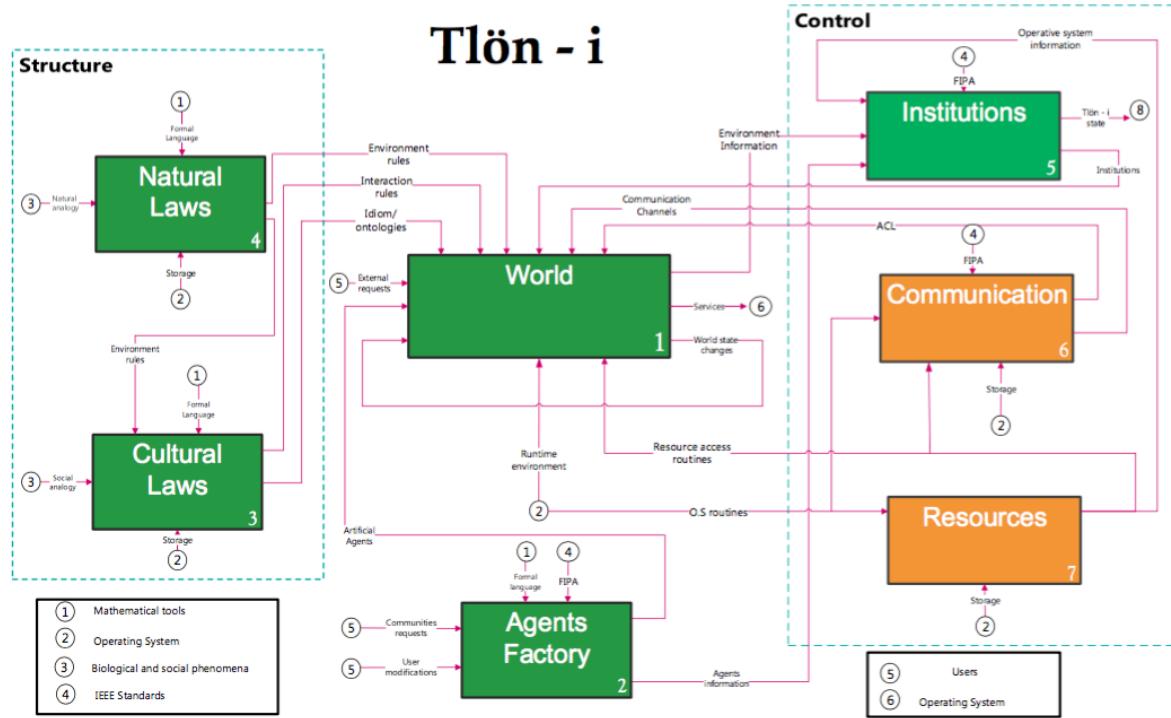


Figure 5-5.: Second level of analysis with SADT

behavior represented a set of threads. All agents encapsulate their internal states and behaviors. Note that this is one of the main differences between an object and an agent from the implementation perspective. Objects only encapsulate their internal state; encapsulating the agent's behavior allows us to consider them as autonomous entities.

The *Resources* component serves as a communication interface between agents and the available computational resources in the system. The idea is to implement a set of mechanisms to allow the agents to use the distributed resources provided by the virtualization layer described in chapter 3. Also, since we expect high levels of heterogeneity, the agents need a common framework to help them to interact and exchange information. Thus, a common ontology and a well-defined interaction format (Protocol) are required to ensure that the messages will have the same meaning for all participants. Accordingly, we adopt a declarative approach used in The FIPA Agent Communication Language (FIPA-ACL) to provide these functionalities through the *Communication* component. Also, the *Register* module has the responsibility to identify, register, and manage the agents of the system. It works as a directory in which the agents are registered and describe services and tasks that they can perform. A detail description of the Register module can be found in [6].

Furthermore, the *Cultural laws* component is responsible for defining the possible interaction

patterns in the system. We explored classical models of game theory like non-cooperative and coalitional games. Some experimental results of this analysis can be found in [4, 3]. Also, this component contains the implementations of social institutions and computational justice based on the model presented in chapter 4. Finally, we have the *World* component that represents an action situation in which the agents and the institutions interact in order to manage and share common-pool resources. We deployed these seven components in each node as an instance of the multi-agent system. We aim to produce the required computational mechanisms to allow the network to operate as a self-organizing system in which the global patterns of behaviors and main functionalities emerge as the result of local interactions.

5.5. Experimental results and performance evaluation

In this Section, we evaluate the multi-agent system through three different scenarios. We present two of the most common application of the ad hoc networks using the agent paradigm to control and regulate the system. First, we explore coverage expansion, which is a promising opportunity due to the capacity of ad hoc networks to operate with several topologies and mobility patterns. Second, we set a sensor network in a rural area in order to collect information regarding the levels of temperature and humidity. This experiment has the purpose of exploring future applications of this proposal in the context of crops and agriculture. Finally, we configured a scenario to evaluate the performance of the computational justice model presented in chapter 4. In all scenarios, we use a set of raspberry pi devices to create an ad hoc network using the IEEE 802.11n. Table 5-1 present the principal hardware features of the devices involved in the experiments. The software characteristics are presented below:

- Linux kernel: Linux 4.9
- Batctl: Batman-adv 2016.4
- A.L.F.R.E.D: Alfred-2016.4
- Multi-agent system: python 3.6

In all experiments, we use the routing protocol B.A.T.M.A.N. (*Better Approach To Mobile Ad hoc Networks*), which is developed by the Freifunk community to replace OLSR protocol. B.A.T.M.A.N. is available in the Linux kernel with some extra functionalities like the A.L.F.R.E.D. Deamon (*Almighty Lightweight Fact Remote Exchange Daemon*) which allow us to disseminate information in the system using a hierarchical configuration. B.A.T.M.A.N. is used at the network layer as a mechanism to exchange information and allow the multi-agent system to communicate different environments at the same time. A detail description of this protocol can be found in [155].

Table 5-1.: Hardware features of the nodes

	Raspeberry pi-2	Raspberry pi-3
CPU	ARM Cortex A7	ARM Cortex A53
CPU speed	900 Mhz	1200 Mhz
RAM	1 GB	1GB
Storage	MicroSDHC port	MicroSDHC port
WI-FI	No built-in Wifi	802.11n

5.5.1. Coverage expansion

A remarkable application of an ad hoc network is its capacity to provide coverage expansion using the system capabilities to operate under different topologies and mobility patterns. For instance, in rural sites or in construction areas in which it could be too expensive or inconvenient to set up a network just for temporary use. Similarly, a related situation occurs when it is commonly needed to relocate network services to one place to other avoiding technical complications. In these scenarios, we can use the natural flexibility provided by self-organizing networks and the agent paradigm to configure a network with these properties. In this regard, using the functionalities provided by the B.A.T.M.A.N. protocol, we can define a node as a gateway that will be responsible for providing internet service. Figure 5-6 presents the network topology. Reactive and reflex agents are used to provide internet service through the multi-agent system. Each node has an instance of the environment and a set of agents to measure and control the system.

During the network operation, we performed connectivity tests sending a ping from each node to the internet provider while the distance between the nodes was increased gradually. We aim to evaluate the network behavior using the throughput and the time intervals in the ping request as performance metrics. Figure 5-7 presents the obtained results. It is possible to notice an increment in the connection times as a result of a dynamic topology. The more distance between the nodes, the more time is needed for establishing a connection. Although it is a natural result, it is important to note the high performance of the routing protocol. It can support periods in which some nodes are not available for sending messages. However, when they return to the connection area, the routing algorithm automatically integrates them into the network. According to the results, the average distance between nodes to keep the network operating in acceptable conditions was the 60 meter. Even when this property is not because of the multi-agent system, it provides a certain level of reliability and robustness for the applications that will be work on it.

For the experiment, we analyzed more than 600 samples in which each sample represent a

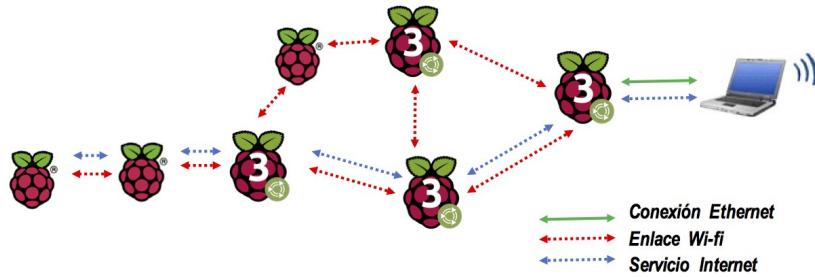


Figure 5-6.: Coverage expansion: network topology

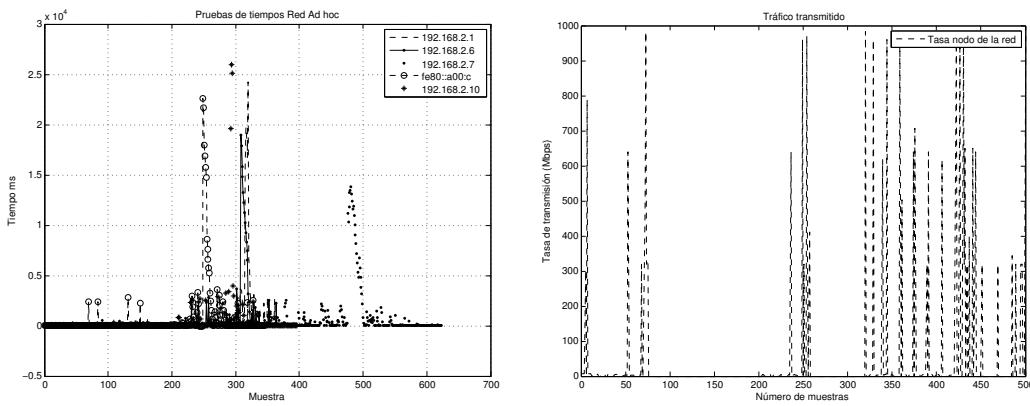


Figure 5-7.: Coverage expansion: performance evaluation

RTT (*round-trip time o round-trip delay time*) measurement. The time responses fluctuate according to the distance between the node and the gateway. Also, Figure 5-7 shows how, for intervals greater than 2.5 seconds, there was an entire disconnection of the network [2]. These results showed the strong influence of mobility patterns and the distance among the nodes in the network performance. Although the nodes kept connected during almost all experiments, the delay increased significantly, producing a restriction over the possible future applications. A detailed description of coverage expansion results can be found in [2].

5.5.2. Sensor network

Sensor networks are used for collecting and processing data in environments without infrastructure [48, 57]. They can be used for environmental monitoring, emergency networks, and smart cities. All these scenarios include multi-hop topologies, resource constraints, and different mobility patterns. Sensor networks are characterized by having considerable limitations regarding computational resources and the control mechanisms for distributing the task in the network. Also, there is a growing research area related to precision agriculture in which wireless technologies are used to observing, measuring, and responding effectively to changes in crops and production systems. It is essential to mention that in Colombia, this is one of

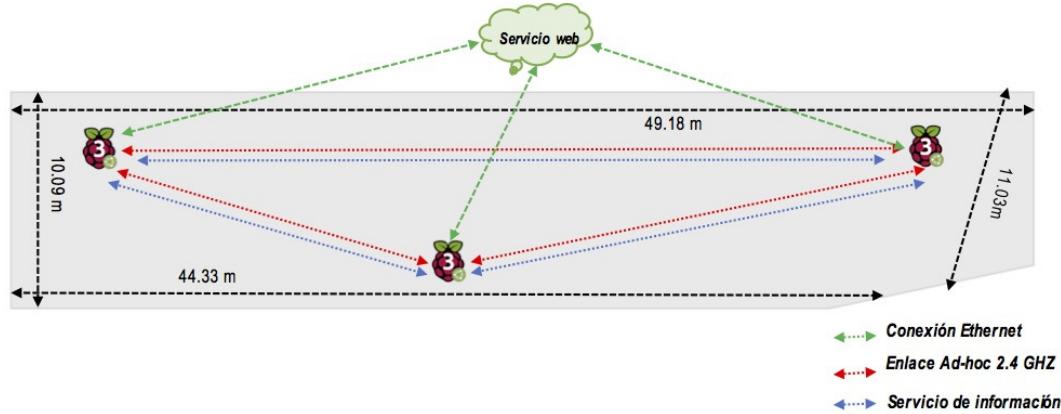


Figure 5-8.: Sensor networks: topology

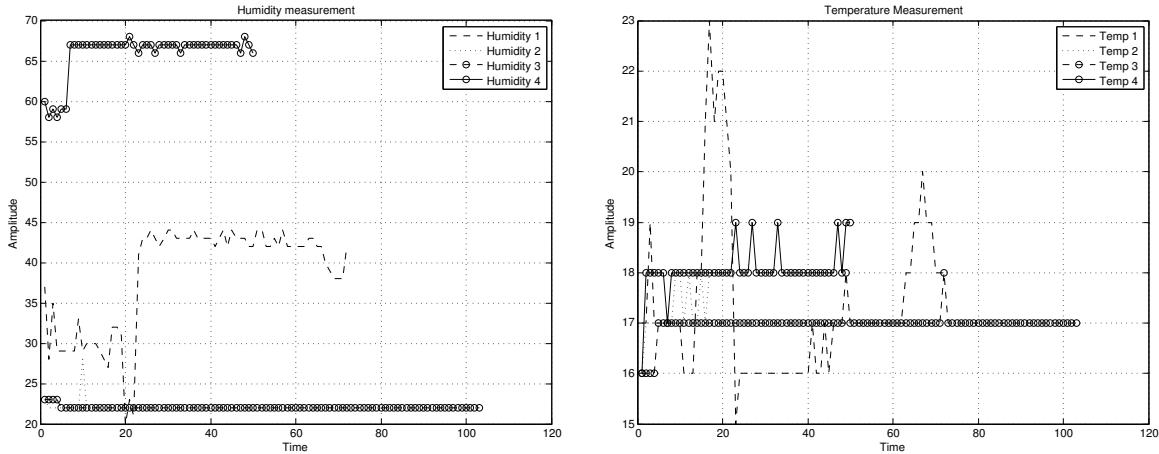


Figure 5-9.: Sensor networks: performance evaluation

the most promising applications of this type of technology. In the long-term, this proposal has the intention of providing a framework for developing these applications using the agent paradigm, ad hoc networks, and socially inspired computing.

In this experiment, we configured a sensor network using the multi-agent system in an area available for farming to evaluate variations in its levels temperature and humidity. The ground has an area of 471.7 m^2 and was located in Sopo, Cundinamarca. Figure 5-6 5-8 describes the network topology. We use reactive and reflex agents to collect and process the data provided by the sensors. Each node has an instance of the environment and a set of agents to control the system. Also, a web service is used to process the data and determine the location of each node using a GPS. Figure 5-9 presents the obtained results. During the experiment, the sensors are sensing the environment and flooding the network with data. The agents are responsible for sending the collected data to the web service. We analyzed

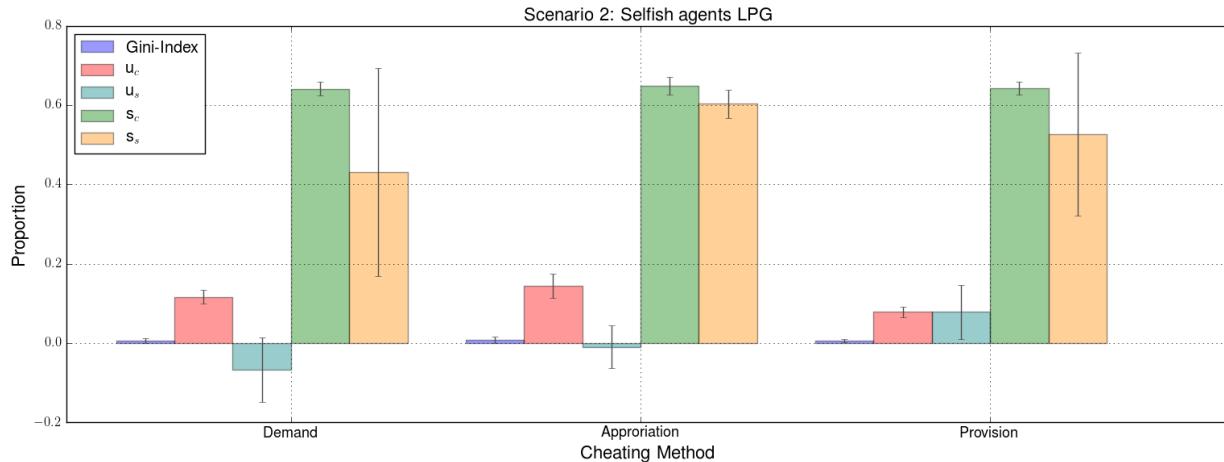


Figure 5-10.: Mobile clouds: computational justice evaluation

400 samples during the experiment. The results showed the direct relationship between the mobility pattern and the performance of the system. However, the information provided by the sensors is analyzed and processed as we expected. It is important to mention that the purpose of this scenario is to explore the future application of this proposal in the context of precision agriculture using low-cost hardware and the flexibility provided agent paradigm.

5.5.3. Mobile clouds

In this scenario, we emulated a mobile cloud to evaluate the computational justice model proposed in chapter 4. In this experiment, all agents have the possibility of sharing their own storage in order to create a public good that can be used for all participants. Also, using distributed storage is an opportunity to deal with possible server failures and take advantage of the free memory in the network devices. All agents' interactions are based on the linear public good game. Also, a self-organizing institution was created for dealing with the distribution problem according to the principles of computational justice. We used the same performance metrics as proposed in chapter 4. Additionally, selfish agents were created to evaluate how they affect the performance of the system regarding cheating behaviors in demand, provision, and allocation actions. Figure 5-10 presents the obtained results. It is essential to mention that due to hardware limitations, this experiment considers only 5 agents and 50 rounds in each game.

The results show how the behavior of the computational justice model in the ad hoc network is very close to the behavior obtained in the simulations presented in chapter 4. Using principles of distributive justice in the context of mobile clouds promotes cooperation prioritizing the allocation over compliant agents and gradually forcing selfish agents to leave the

system. Also, scenarios composed of cheating appropriators show the ineffectiveness of this model to deal with this selfish strategy. [60]. In general terms, these results agree with the expected behavior and allow us to consider the use of this proposal for dealing with the distribution problem in the context of mobile clouds. It is important to mention that even when the model presents an acceptable behavior, there are hardware constraints that we need to consider if we want to implement this model in a real environment. In particular, delays in communication and the aggregate computational cost of the distributed resources operation.

5.6. Conclusions

In this chapter, a multi-agent system for self-organizing communication networks was proposed. We overview some of the most significant applications of the ad hoc networks and their connection with the philosophy behind the agent paradigm. Two different approaches for creating artificial agents were presented: first, the functional approach that allows us to model and implement different types of behavior based on the observed actions of the agents. Second, the Sartrean model, which provides us a structural perspective based on biologically and socially inspired computing. A proposed architecture for the multi-agent system was described using the *Structured analysis and design technique* (SADT) in which the main components of the system were explained. Furthermore, we explored the performance of the system through three different experimental scenarios in which we evaluate applications like coverage expansion, sensor network, and the application of computational justice in the context of mobile clouds. Despite these applications belong to an exploratory stage of this research, they show the relevance of the socially inspiring computing in the context of communication systems; in particular, ad hoc networks.

6. Conclusions

We showed that it is possible to use biologically and socially inspired computing for building communications systems. We argue that an abstract analysis of biological and social phenomena can be made to create a conceptual framework for developing a new kind of networking technology. Biologically inspired computing can be used for achieving efficient and scalable networking under uncertain conditions and socially inspired computing for solving problems through collective actions. The combination of these two approaches enables us to develop communication networks not only enough robust and adaptive to operate in highly dynamic environments but also with the capacity to use collective behaviors for solving complex problems.

Additionally, the current fairness and computational justice models for communication systems were overviewed and compared. We presented two different approaches: first, the classical engineering approach, which usually uses a set of measures (quantitative or qualitative) to assess a resource allocation according to a set of parameters. Although there are a considerable number of proposals related to this approach, most of them need a centralized operation and complete information of the system to provide a fairness analysis. Second, socially inspired computing approach which uses social concepts like Institutions, Distributive Justice, and cooperation to create allocation methods based on the behavior of human society. Despite there is no direct application of socially inspiring computing in communication systems, they provide a set of principles that can be used to face the distribution problem in self-organizing communication networks; in particular, ad hoc networks.

We proved that it is possible to deal with the distribution problem in the self-organizing open system using a combination of socially inspired computing and agent-based modeling. In particular, this proposal was composed of: the Ostrom's approach for institutional analysis, cooperation patterns through stochastic games, the Rescher's idea of distributive justice, and adaptive mechanisms through voting systems and genetic algorithms. The result showed how an allocation process based on computational justice is a potential alternative to deal with the distribution problem in environments formed by autonomous agents operating without a central controller or other orchestration forms. It is important to mention that this work was developed on the foundation of self-organizing electronic institutions established

by Pitt et al. [1], in which they showed the potential value of the distributive justice and social institutions as part of the future technological developments.

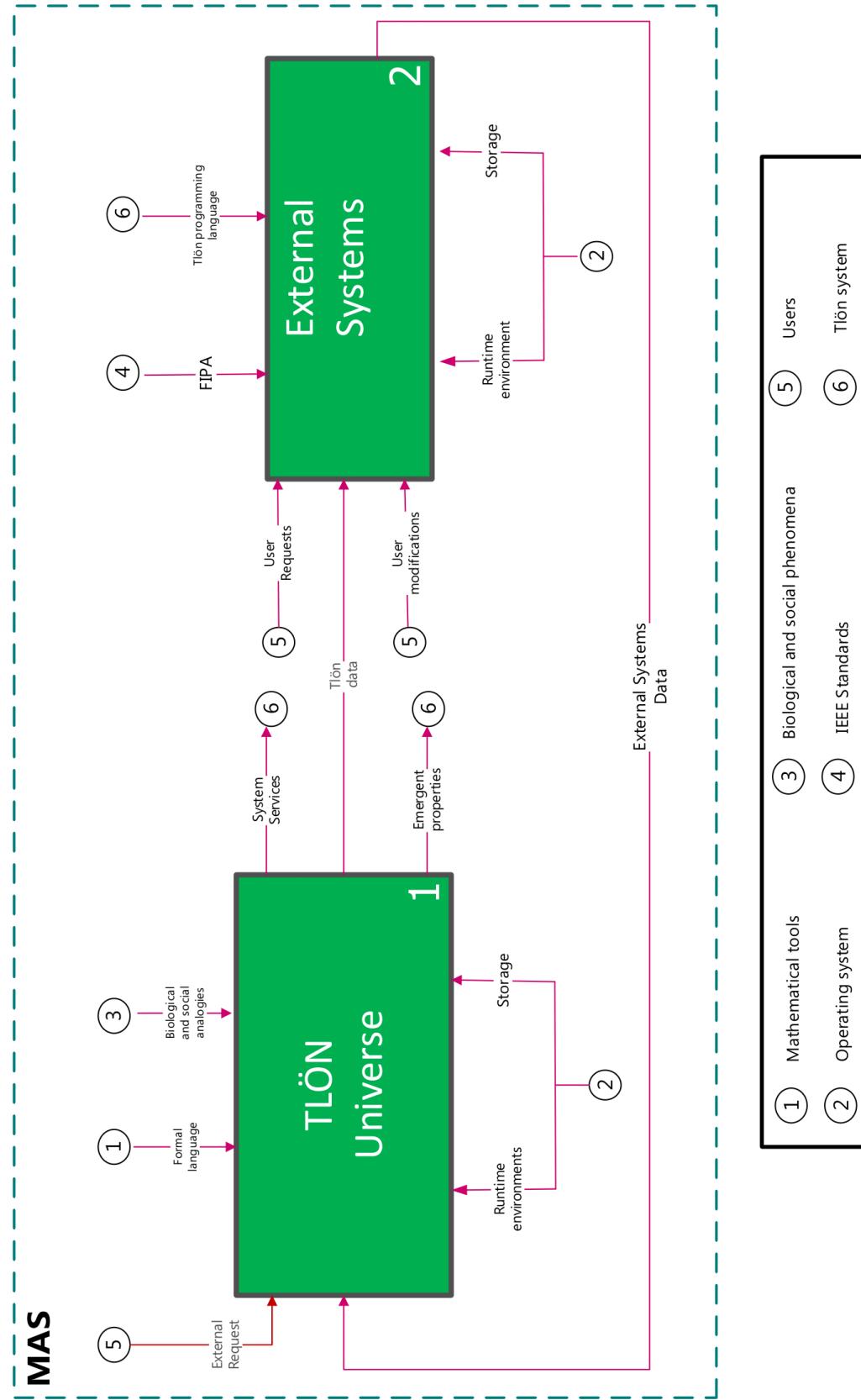
Furthermore, the notion of action situation allows us to maintain the structure of the multi-agent system no matter if the cooperation pattern, the allocation method, the adaptive mechanisms, or the voting system are changed. This structure enables us to do a meta-analysis of the distribution problem not only in terms of the main components involved in the model but also regarding the information flows in the system. Moreover, the inclusion of stochastic games as part of the cooperation patterns results in a suitable choice for modeling the relationship between the environment state and the agents' actions; this feature gives us a more realistic version of the distribution problem and results in better cooperation levels. Moreover, we modified the canon of equality introduced in the initial proposal of computational justice, producing a distribution method that can deal with cheating actions in provision and demand without the use of adaptive strategies; this feature decreases the computational cost involved in the allocation process and represents an improvement regarding previous works.

Finally, a multi-agent system for self-organizing communication networks was proposed. We overview some of the most significant applications of the ad hoc networks and their connection with the philosophy behind the agent paradigm. Two different approaches for creating artificial agents were presented: first, the functional approach that allows us to model and implement different types of behavior based on the observed actions of the agents. Second, the Sartrean model, which provides us a structural perspective based on biologically and socially inspired computing. A proposed architecture for the multi-agent system was described using the *Structured analysis and design technique* (SADT) in which the main components of the system were explained. Furthermore, we explored the performance of the system through three different experimental scenarios in which we evaluate applications like coverage expansion, sensor network, and the application of computational justice in the context of mobile clouds. Although these applications belong to an exploratory stage of this research, they show the relevance of the socially inspiring computing in the context of communication systems.

A. Appendix A: MAS architecture using SADT

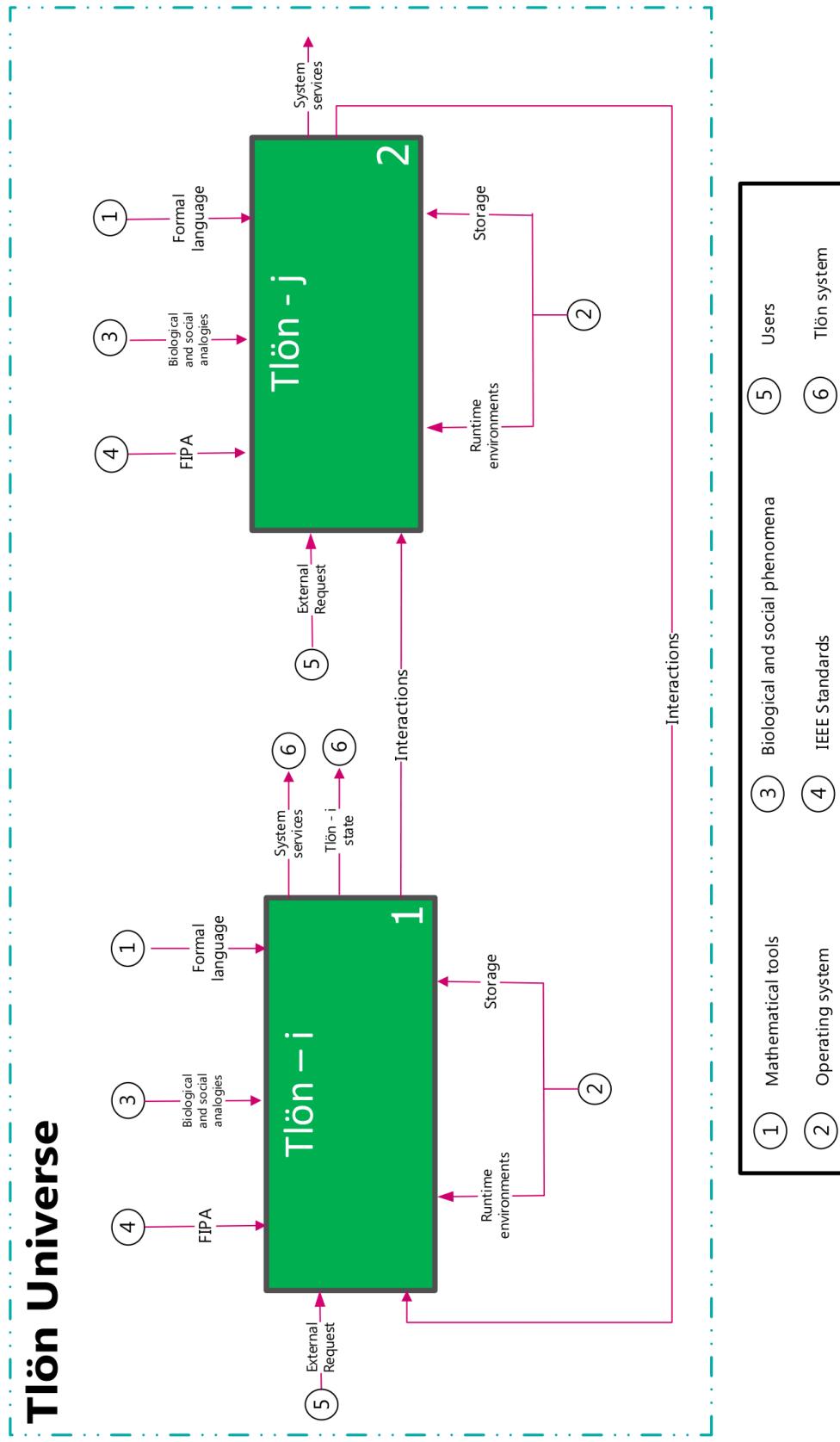
In this section, we present the different levels of the multi-agent system using the SADT methodology. A detail description of this proposal and the full source code are available for download at https://github.com/jospinalo/MAS_TLON.

Multi-agent system

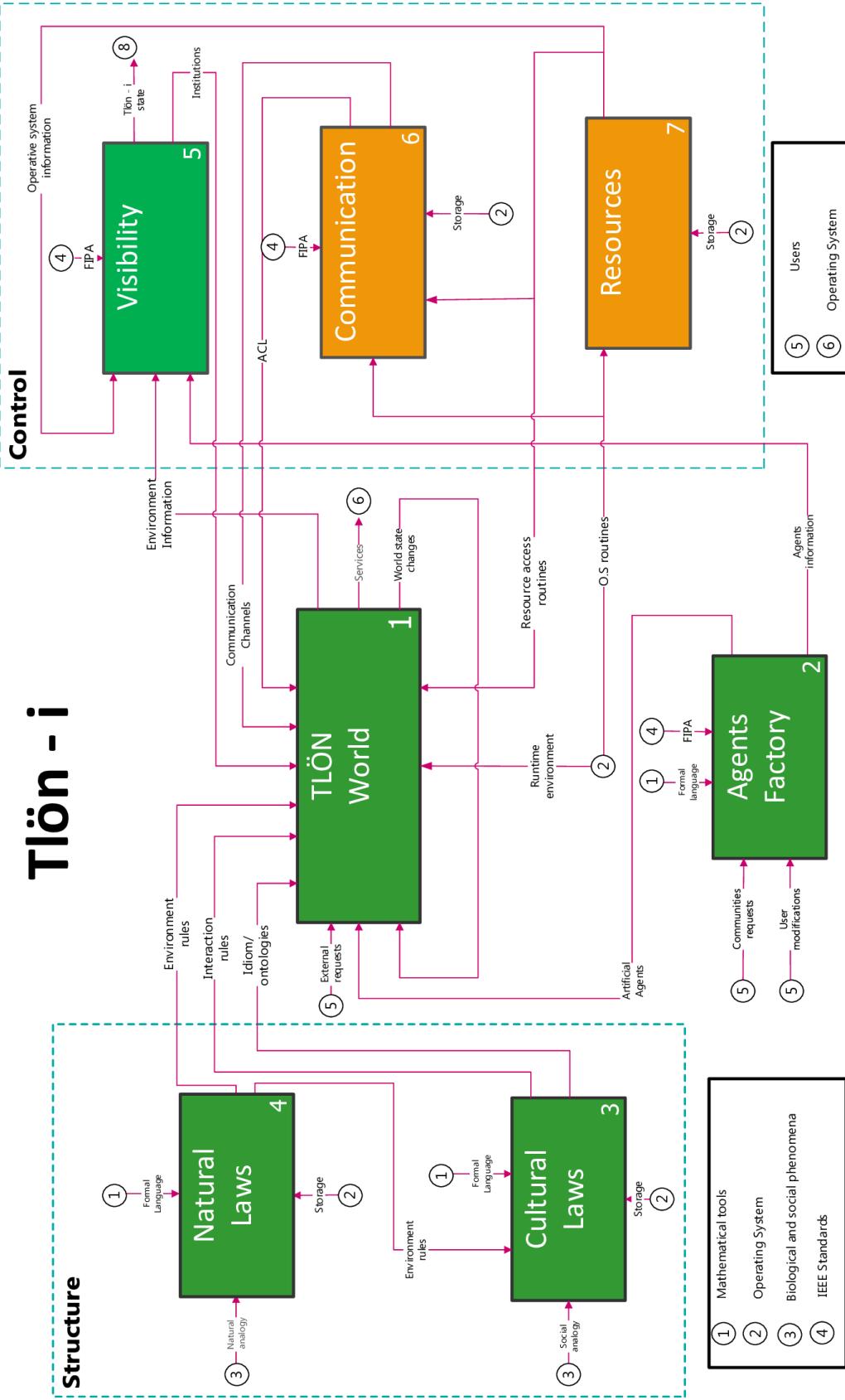


Interactions network

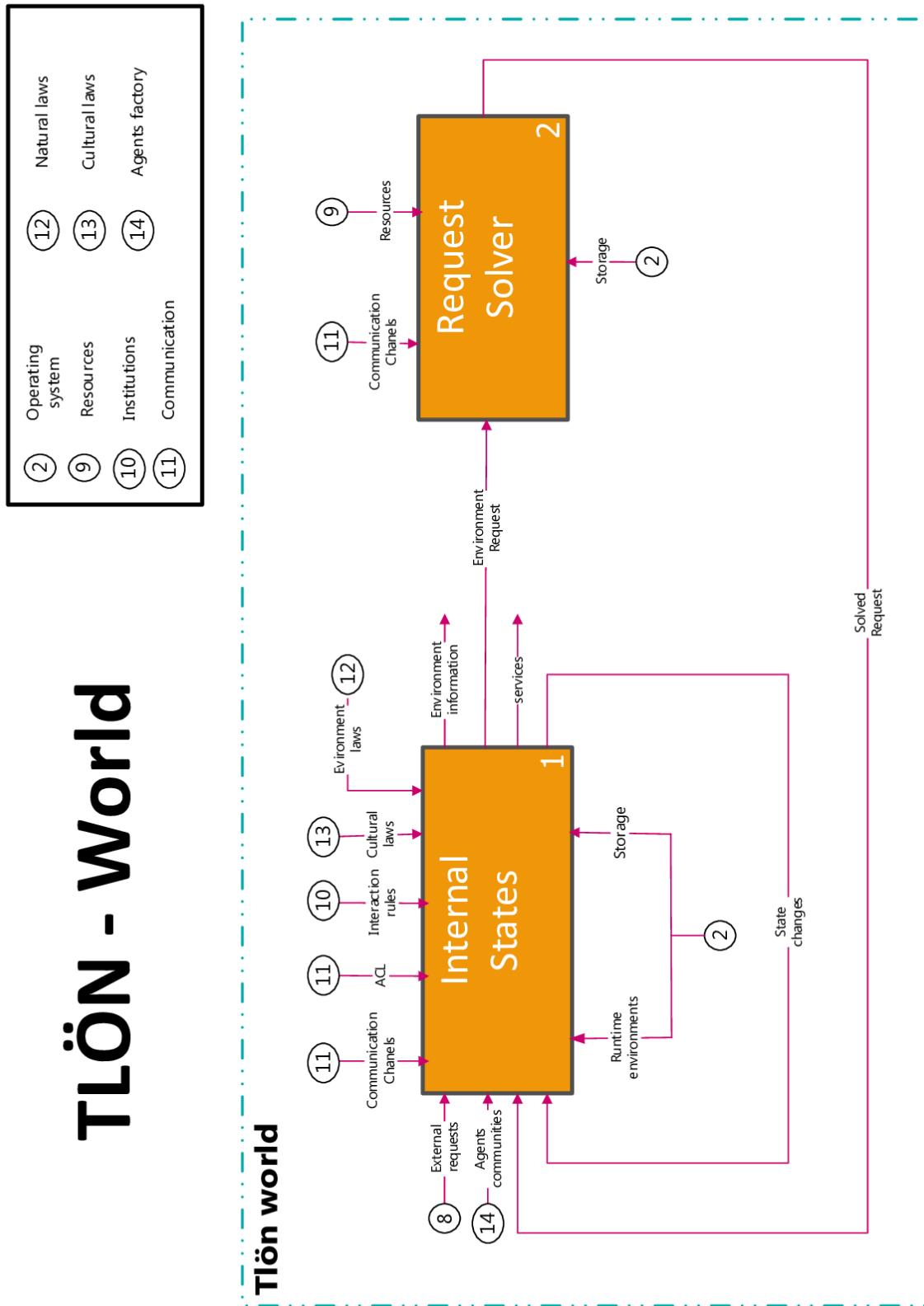
Tlön Universe



Tlön - i

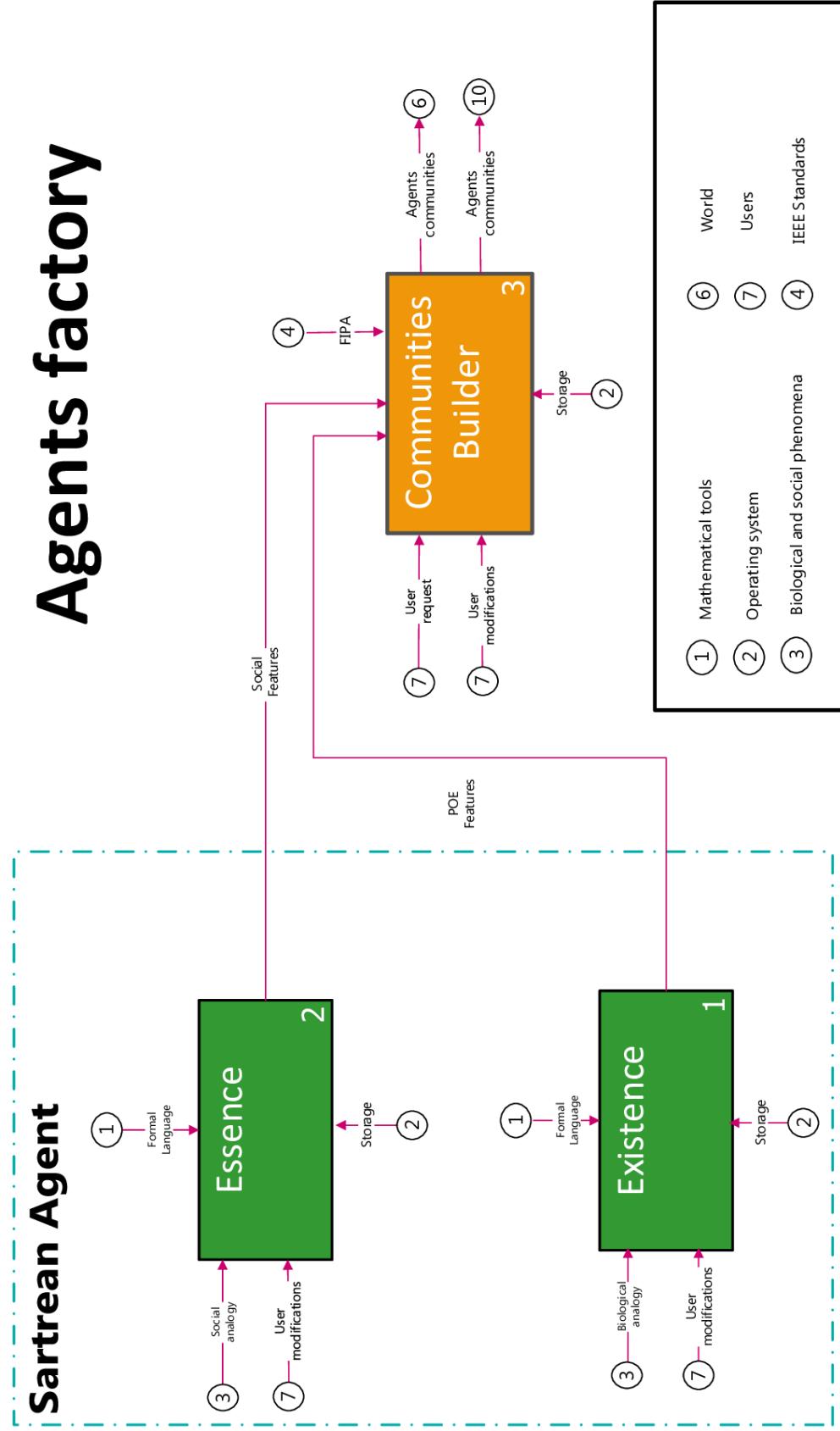


TLÖN - World

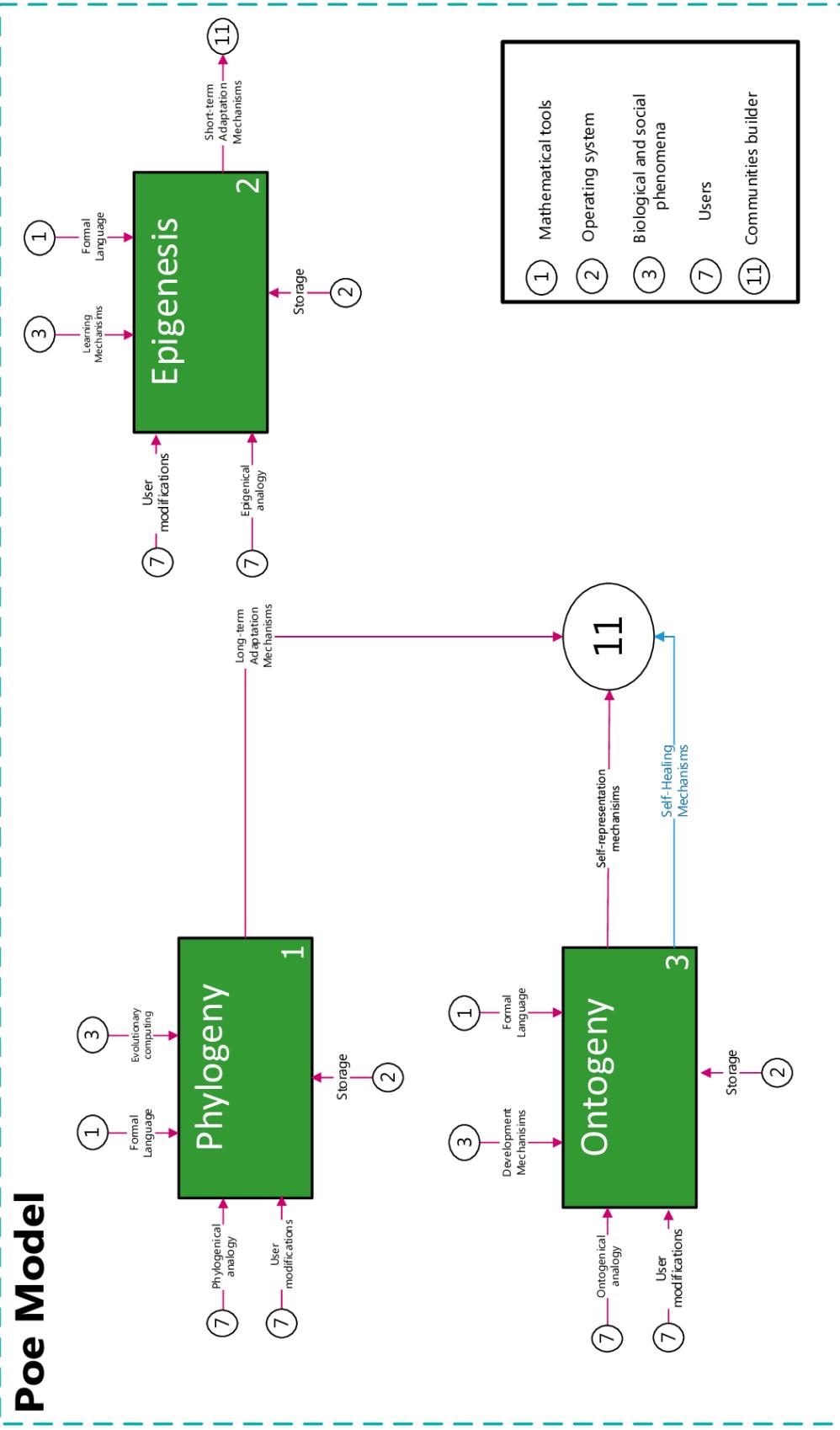


Sartrean Agent

Agents factory

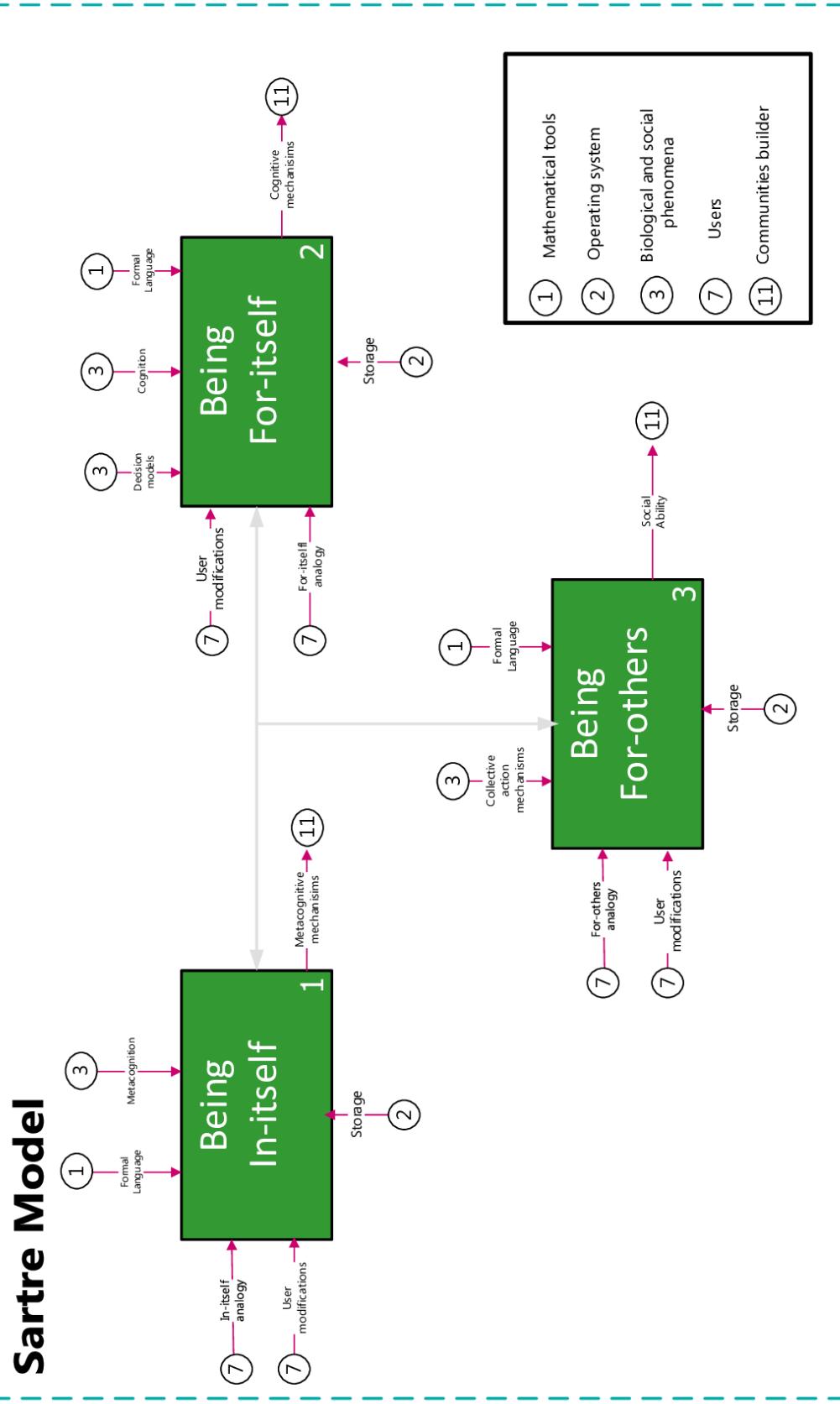


Existence

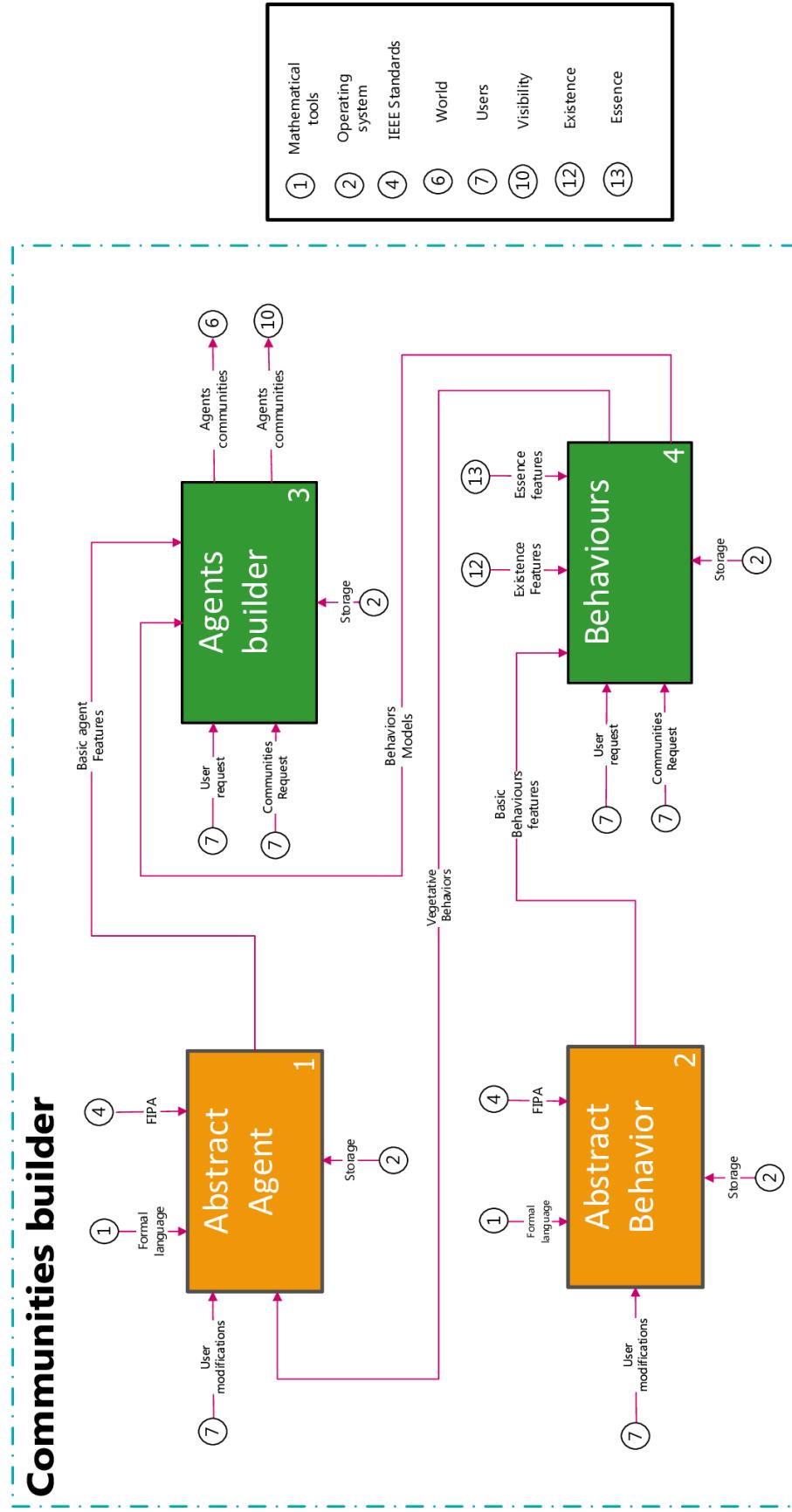


Essence

Sartre Model



Communities builder



Bibliography

- [1] J. Pitt, D. Busquets, and S. Macbeth, “Distributive justice for self-organised common-pool resource management,” *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 9, no. 3, p. 14, 2014.
- [2] J. F. Sánchez, J. P. Ospina, J. Lopez, and J. Ortiz, “Proposed ad hoc network to expand coverage of an ieee 802.11 n network,” in *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2017, pp. 1–5.
- [3] D. A. Vega, J. P. Ospina, J. F. Latorre, and J. E. Ortiz, “An adaptive trust model for achieving emergent cooperation in ad hoc networks,” in *Current Trends in Semantic Web Technologies: Theory and Practice*. Springer, 2019, pp. 85–100.
- [4] J. F. Latorre, J. P. Ospina, and J. E. Ortiz, “A coalitional game for achieving emergent cooperation in ad hoc networks through sympathy and commitment,” in *Workshop on Engineering Applications*. Springer, 2018, pp. 352–362.
- [5] J. P. Ospina and J. E. Ortiz, “Estimation of a growth factor to achieve scalable ad hoc networks,” *Ingeniería y Universidad*, vol. 21, no. 1, pp. 49–70, 2017.
- [6] L. J. Alfonso, J. P. Ospina, and J. E. Ortiz, “A register module for agents communities through robustness property in ad hoc networks,” in *2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*. IEEE, 2018, pp. 1–6.
- [7] J. F. Sánchez, J. P. Ospina, J. E. Gonzalez, and J. E. Ortiz, “Sociability as a decision model for artificial agents,” in *Proceedings of the 2017 International Symposium on Industrial Engineering and Operations Management (IEOM) IEOM Society*, 2017.
- [8] G. R. Osorio, J. F. Sánchez, J. P. Ospina, and J. E. Ortiz, “An address allocation protocol for ad hoc networks through pollen dispersion algorithms,” in *2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI)*. IEEE, 2019, pp. 1–6.
- [9] J. P. Ospina, J. F. Sánchez, J. E. Ortiz, C. Collazos-Morales, and P. Ariza-Colpas, “Socially and biologically inspired computing for self-organizing communications net-

- works,” in *International Conference on Machine Learning for Networking*. Springer, 2019, pp. 461–484.
- [10] J. D. Day and H. Zimmermann, “The osi reference model,” *Proceedings of the IEEE*, vol. 71, no. 12, pp. 1334–1340, 1983.
 - [11] T. G. Robertazzi, *Computer networks and systems: queueing theory and performance evaluation*. Springer Science & Business Media, 2012.
 - [12] V. B. Iversen, “Teletraffic engineering and network planning,” *Technical University of Denmark*, p. 270, 2010.
 - [13] F. Dressler, “A study of self-organization mechanisms in ad hoc and sensor networks,” *Computer Communications*, vol. 31, no. 13, pp. 3018–3029, 2008.
 - [14] T. Qiu, N. Chen, K. Li, D. Qiao, and Z. Fu, “Heterogeneous ad hoc networks: Architectures, advances and challenges,” *Ad Hoc Networks*, vol. 55, pp. 143–152, 2017.
 - [15] V. Pureswaran and P. Brody, “Device democracy: Saving the future of the internet of things,” *IBM Corporation*, 2015.
 - [16] C. Gershenson and F. Heylighen, “When can we call a system self-organizing?” in *European Conference on Artificial Life*. Springer, 2003, pp. 606–614.
 - [17] C. E. Maldonado, “Complejidad: ciencia, pensamiento y aplicaciones,” *Books*, vol. 1, 2007.
 - [18] F. Dressler, *Self-organization in sensor and actor networks*. John Wiley & Sons, 2008.
 - [19] M. S. Taqqu, W. Willinger, and R. Sherman, “Proof of a fundamental result in self-similar traffic modeling,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 5–23, 1997.
 - [20] M. Alzate and A. Monroy, “Introducción al tráfico autosimilar en redes de comunicaciones,” *Revista Ingeniería*, vol. 6, no. 2, pp. 6–17, 2001.
 - [21] P. Ranjan, E. H. Abed, and R. J. La, “Nonlinear instabilities in tcp-red,” *IEEE/ACM transactions on networking*, vol. 12, no. 6, pp. 1079–1092, 2004.
 - [22] S. Mukherjee, R. Ray, R. Samanta, M. H. Khondekar, and G. Sanyal, “Nonlinearity and chaos in wireless network traffic,” *Chaos, Solitons & Fractals*, vol. 96, pp. 23–29, 2017.

- [23] W. San-Um, P. Ketthong, and J. Noymanee, “A deterministic node mobility model for mobile ad hoc wireless network using signum-based discrete-time chaotic map,” in *Telecommunication Networks and Applications Conference (ITNAC), 2015 International*. IEEE, 2015, pp. 114–119.
- [24] B. Vaseghi, M. A. Pourmina, and S. Mobayen, “Secure communication in wireless sensor networks based on chaos synchronization using adaptive sliding mode control,” *Nonlinear Dynamics*, vol. 89, no. 3, pp. 1689–1704, 2017.
- [25] S. Wang, Y. Chen, and H. Tian, “An intrusion detection algorithm based on chaos theory for selecting the detection window size,” in *Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on*. IEEE, 2016, pp. 556–560.
- [26] L. Kocarev, *Complex dynamics in communication networks*. Springer Science & Business Media, 2005.
- [27] A.-L. Barabási and E. Bonabeau, “Scale-free networks,” *Scientific american*, vol. 288, no. 5, pp. 60–69, 2003.
- [28] L. Li, D. Alderson, W. Willinger, and J. Doyle, “A first-principles approach to understanding the internet’s router-level topology,” in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 3–14.
- [29] M. Mihail, C. Papadimitriou, and A. Saberi, “On certain connectivity properties of the internet topology,” *Journal of Computer and System Sciences*, vol. 72, no. 2, pp. 239 – 251, 2006, jCSS FOCS 2003 Special Issue.
- [30] F. Dressler *et al.*, “Self-organization in ad hoc networks: Overview and classification,” *University of Erlangen, Dept. of Computer Science*, vol. 7, pp. 1–12, 2006.
- [31] C. Gershenson, *Design and control of self-organizing systems*. CopIt ArXives, 2007.
- [32] A. S. Tanenbaum and M. Van Steen, *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.
- [33] D. Abadi, “Consistency tradeoffs in modern distributed database system design: Cap is only part of the story,” *Computer*, vol. 45, no. 2, pp. 37–42, 2012.
- [34] R. Smith, *Docker Orchestration*. Packt Publishing Ltd, 2017.
- [35] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*. John Wiley & Sons, 2007, vol. 7.

- [36] B. C. Neuman, “Scale in distributed systems,” *ISI/USC*, 1994.
- [37] F. Heylighen and C. Gershenson, “The meaning of self-organization in computing,” *IEEE Intelligent Systems*, vol. 18, no. 4, 2003.
- [38] J. Pitt, A. Diaconescu, and A. Bourazeri, “Democratisation of the smartgrid and the active participation of prosumers,” in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1707–1714.
- [39] J. Loo, J. L. Mauri, and J. H. Ortiz, *Mobile ad hoc networks: current status and future trends*. CRC Press, 2016.
- [40] G. Carreón, C. Gershenson, and L. Pineda, “Improving public transportation systems with self-organization: A headway-based model and regulation of passenger alighting and boarding.” *PloS one*, vol. 12, no. 12, 2017.
- [41] P. Grippa, D. A. Behrens, C. Bettstetter, and F. Wall, “Job selection in a network of autonomous uavs for delivery of goods,” *arXiv preprint arXiv:1604.04180*, 2016.
- [42] F. Dressler, F. Klingler, C. Sommer, and R. Cohen, “Not all vanet broadcasts are the same: Context-aware class based broadcast,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 17–30, 2018.
- [43] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, “Drone networks: Communications, coordination, and sensing,” *Ad Hoc Networks*, vol. 68, pp. 1–15, 2018.
- [44] S. Li, L. Da Xu, and S. Zhao, “5g internet of things: A survey,” *Journal of Industrial Information Integration*, 2018.
- [45] P. Dunphy and F. A. Petitcolas, “A first look at identity management schemes on the blockchain,” *arXiv preprint arXiv:1801.03294*, 2018.
- [46] J. Ramiro and K. Hamied, *Self-organizing networks (SON): Self-planning, self-optimization and self-healing for GSM, UMTS and LTE*. John Wiley & Sons, 2011.
- [47] F. Dressler and O. B. Akan, “A survey on bio-inspired networking,” *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.
- [48] K. Yang, “Wireless sensor networks,” *Principles, Design and Applications*, 2014.
- [49] J. Pitt, J. Schaumeier, and A. Artikis, “The axiomatisation of socio-economic principles for self-organising systems,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*. IEEE, 2011, pp. 138–147.

- [50] T. Tsuchiya and T. Kikuno, “An adaptive mechanism for epidemic communication,” in *International Workshop on Biologically Inspired Approaches to Advanced Information Technology*. Springer, 2004, pp. 306–316.
- [51] M. Dorigo and M. Birattari, “Ant colony optimization,” in *Encyclopedia of machine learning*. Springer, 2011, pp. 36–39.
- [52] M. Mejia, N. Peña, J. L. Muñoz, O. Esparza, and M. A. Alzate, “A game theoretic trust model for on-line distributed evolution of cooperation inmanets,” *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 39–51, 2011.
- [53] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [54] Y. Zhang, J. Zheng, and H.-H. Chen, *Cognitive radio networks: architectures, protocols, and standards*. CRC press, 2016.
- [55] A. Galati, “Delay tolerant network,” 2010.
- [56] A. J. Jones, A. Artikis, and J. Pitt, “The design of intelligent socio-technical systems,” *Artificial Intelligence Review*, vol. 39, no. 1, pp. 5–20, 2013.
- [57] F. Wortmann and K. Flüchter, “Internet of things,” *Business & Information Systems Engineering*, vol. 57, no. 3, pp. 221–224, 2015.
- [58] P. F. Dell, “Beyond homeostasis: Toward a concept of coherence,” *Family Process*, vol. 21, no. 1, pp. 21–41, 1982.
- [59] J. Pitt, “From trust and forgiveness to social capital and justice: Formal models of social processes in open distributed systems,” in *Trustworthy Open Self-Organising Systems*. Springer, 2016, pp. 185–208.
- [60] J. Pitt, D. Busquets, and R. Riveret, “The pursuit of computational justice in open systems,” *AI & society*, vol. 30, no. 3, pp. 359–378, 2015.
- [61] F. H. Fitzek and M. D. Katz, *Mobile clouds: Exploiting distributed resources in wireless, mobile and social networks*. John Wiley & Sons, 2013.
- [62] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, “Platoon management with cooperative adaptive cruise control enabled by vanet,” *Vehicular communications*, vol. 2, no. 2, pp. 110–123, 2015.

- [63] M. A. Nowak, “Five rules for the evolution of cooperation,” *science*, vol. 314, no. 5805, pp. 1560–1563, 2006.
- [64] J. Pitt, J. Schaumeier, D. Busquets, and S. Macbeth, “Self-organising common-pool resource allocation and canons of distributive justice,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 119–128.
- [65] J. Pitt and J. Schaumeier, “Provision and appropriation of common-pool resources without full disclosure,” in *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 2012, pp. 199–213.
- [66] J. Pitt, A. Diaconescu, and A. Bourazeri, “Democratisation of the smartgrid and the active participation of prosumers,” in *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*. IEEE, 2017, pp. 1707–1714.
- [67] J. P. Garbiso, A. Diaconescu, M. Coupechoux, J. Pitt, and B. Leroy, “Distributive justice for fair auto-adaptive clusters of connected vehicles,” in *Foundations and Applications of Self* Systems (FAS* W), 2017 IEEE 2nd International Workshops on*. IEEE, 2017, pp. 79–84.
- [68] P. E. Petruzzi, J. Pitt, and D. Busquets, “Electronic social capital for self-organising multi-agent systems,” *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 12, no. 3, p. 13, 2017.
- [69] V. Pureswaran and P. Brody, “Device democracy: Saving the future of the internet of things,” *IBM Corporation*, 2015.
- [70] M. Eigen and P. Schuster, *The hypercycle: a principle of natural self-organization*. Springer Science & Business Media, 2012.
- [71] M. Wang and T. Suda, “The bio-networking architecture: A biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications,” in *Applications and the Internet, 2001. Proceedings. 2001 Symposium on*. IEEE, 2001, pp. 43–53.
- [72] U. Hatfield, “Socially inspired computing,” 2005.
- [73] R. Axelrod and W. D. Hamilton, “The evolution of cooperation,” *science*, vol. 211, no. 4489, pp. 1390–1396, 1981.
- [74] A. Videla, “Metaphors we compute by,” *Communications of the ACM*, vol. 60, no. 10, pp. 42–45, 2017.

- [75] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*. McGraw-Hill New York, 2007, vol. 3.
- [76] C. R. Shalizi *et al.*, “Causal architecture, complexity and self-organization in the time series and cellular automata,” Ph.D. dissertation, University of Wisconsin–Madison, 2001.
- [77] A. Giagkos and M. S. Wilson, “Beeip—a swarm intelligence based routing for wireless ad hoc networks,” *Information Sciences*, vol. 265, pp. 23–35, 2014.
- [78] M. Hammoudeh and R. Newman, “Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance,” *Information Fusion*, vol. 22, pp. 3–15, 2015.
- [79] M. Fahad, F. Aadil, S. Khan, P. A. Shah, K. Muhammad, J. Lloret, H. Wang, J. W. Lee, I. Mehmood *et al.*, “Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks,” *Computers & Electrical Engineering*, 2018.
- [80] S. Saleh, M. Ahmed, B. M. Ali, M. F. A. Rasid, and A. Ismail, “A survey on energy awareness mechanisms in routing protocols for wireless sensor networks using optimization methods,” *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 12, pp. 1184–1207, 2014.
- [81] X. Zhang, X. Zhang, and C. Gu, “A micro-artificial bee colony based multicast routing in vehicular ad hoc networks,” *Ad Hoc Networks*, vol. 58, pp. 213–221, 2017.
- [82] G. Di Caro, F. Ducatelle, and L. M. Gambardella, “Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 16, no. 5, pp. 443–455, 2005.
- [83] D. Karaboga and B. Akay, “A survey: algorithms simulating bee swarm intelligence,” *Artificial intelligence review*, vol. 31, no. 1-4, p. 61, 2009.
- [84] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, “Time synchronization in wsns: A maximum-value-based consensus approach,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 660–675, 2014.
- [85] F. Zhang, H. L. Trentelman, and J. M. Scherpen, “Fully distributed robust synchronization of networked lur’e systems with incremental nonlinearities,” *Automatica*, vol. 50, no. 10, pp. 2515–2526, 2014.
- [86] M. Chovanec, J. Milanova, L. Cehovic, P. Sarafin, M. Hudik, and M. Kochlan, “Firefly-based universal synchronization algorithm in wireless sensor network,” in *Advances in Network Systems*. Springer, 2017, pp. 71–86.

- [87] J. He, H. Li, J. Chen, and P. Cheng, "Study of consensus-based time synchronization in wireless sensor networks," *ISA transactions*, vol. 53, no. 2, pp. 347–357, 2014.
- [88] J.-Y. Jung, H.-H. Choi, and J.-R. Lee, "Survey of bio-inspired resource allocation algorithms and mac protocol design based on a bio-inspired algorithm for mobile ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 119–127, 2018.
- [89] G. Brandner, U. Schilcher, and C. Bettstetter, "Firefly synchronization with phase rate equalization and its experimental analysis in wireless systems," *Computer Networks*, vol. 97, pp. 74–87, 2016.
- [90] B. Roh, M.-h. Han, M. Hoh, H.-S. Park, K. Kim, and B.-h. Roh, "Distributed call admission control for desync-tdma in mobile ad hoc networks," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 411–413.
- [91] A.-S. K. Pathan, *Security of self-organizing networks: MANET, WSN, WMN, VANET*. CRC press, 2016.
- [92] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 10, p. 351047, 2013.
- [93] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 266–282, 2014.
- [94] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on knn classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, 2014.
- [95] S. Jamali and R. Fotohi, "Defending against wormhole attack in manet using an artificial immune system," *New Review of Information Networking*, vol. 21, no. 2, pp. 79–100, 2016.
- [96] S. Tiwari, K. Mishra, N. Saxena, N. Singh, and A. Misra, "Artificial immune system based mac layer misbehavior detection in manet," in *Advanced Computer and Communication Engineering Technology*. Springer, 2016, pp. 707–722.
- [97] H. B. Liaqat, F. Xia, Q. Yang, Z. Xu, A. M. Ahmed, and A. Rahim, "Bio-inspired packet dropping for ad-hoc social networks," *International Journal of Communication Systems*, vol. 30, no. 1, 2017.

- [98] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, “Modeling the propagation of worms in networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 942–960, 2014.
- [99] L.-X. Yang, X. Yang, J. Liu, Q. Zhu, and C. Gan, “Epidemics of computer viruses: A complex-network approach,” *Applied Mathematics and Computation*, vol. 219, no. 16, pp. 8705–8717, 2013.
- [100] X. Yang, B. K. Mishra, and Y. Liu, “Computer virus: theory, model, and methods,” *Discrete Dynamics in Nature and Society*, vol. 2012, 2012.
- [101] C. Gan, X. Yang, W. Liu, Q. Zhu, J. Jin, and L. He, “Propagation of computer virus both across the internet and external computers: A complex-network approach,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 8, pp. 2785–2792, 2014.
- [102] B. Liu, W. Zhou, L. Gao, H. Zhou, T. H. Luan, and S. Wen, “Malware propagations in wireless ad hoc networks,” *IEEE Transactions on Dependable and Secure Computing*, 2016.
- [103] M. Chitra and S. S. Sathya, “Bidirectional data dissemination in vehicular ad hoc networks using epidemic spreading model,” in *Proceedings of the International Conference on Informatics and Analytics*. ACM, 2016, p. 57.
- [104] K. Sugihara and N. Hayashibara, “Collecting data in sensor networks using homesick lévy walk,” in *International Conference on Network-Based Information Systems*. Springer, 2017, pp. 779–786.
- [105] Z. Zhang, K. Long, and J. Wang, “Self-organization paradigms and optimization approaches for cognitive radio technologies: a survey,” *IEEE Wireless Communications*, vol. 20, no. 2, pp. 36–42, 2013.
- [106] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, “Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.
- [107] Z. Zhang, K. Long, J. Wang, and F. Dressler, “On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 513–537, 2014.

- [108] Z. Zhang, W. Huangfu, K. Long, X. Zhang, X. Liu, and B. Zhong, “On the designing principles and optimization approaches of bio-inspired self-organized network: a survey,” *Science China Information Sciences*, vol. 56, no. 7, pp. 1–28, 2013.
- [109] D. G. Reina, P. Ruiz, R. Ciobanu, S. Toral, B. Dorronsoro, and C. Dobre, “A survey on the application of evolutionary algorithms for mobile multihop ad hoc network optimization problems,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 2, p. 2082496, 2016.
- [110] S. N. Meera, D. S. Kumar, and S. Rao, “Ad hoc networks: Route discovery channel for mobile network with low power consumption,” in *Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications*. Springer, 2018, pp. 665–671.
- [111] F. Dressler and O. B. Akan, “Bio-inspired networking: from theory to practice,” *IEEE Communications Magazine*, vol. 48, no. 11, 2010.
- [112] K. Leibnitz, N. Wakamiya, and M. Murata, “Biologically inspired networking,” *Cognitive Networks: Towards Self-Aware Networks*, pp. 1–21, 2007.
- [113] D. Câmara, *Bio-Inspired Networking*. Elsevier, 2015.
- [114] N. Pazhaniraja, P. V. Paul, G. Roja, K. Shanmugapriya, and B. Sonali, “A study on recent bio-inspired optimization algorithms,” in *Signal Processing, Communication and Networking (ICSCN), 2017 Fourth International Conference on*. IEEE, 2017, pp. 1–6.
- [115] M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Pérez-Uribe, and A. Stauffer, “A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 83–97, 1997.
- [116] R. Sugumar, A. Rengarajan, and C. Jayakumar, “Trust based authentication technique for cluster based vehicular ad hoc networks (vanet),” *Wireless Networks*, vol. 24, no. 2, pp. 373–382, 2018.
- [117] M. Mejia, N. Pena, J. L. Munoz, and O. Esparza, “A review of trust modeling in ad hoc networks,” *Internet Research*, vol. 19, no. 1, pp. 88–104, 2009.
- [118] Y. Li, H. Xu, Q. Cao, Z. Li, and S. Shen, “Evolutionary game-based trust strategy adjustment among nodes in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 2, p. 818903, 2015.
- [119] L. A. Grieco, A. Rizzo, S. Colucci, S. Sicari, G. Piro, D. Di Paola, and G. Boggia, “Iot-aided robotics applications: Technological implications, target domains and open issues,” *Computer Communications*, vol. 54, pp. 32–47, 2014.

- [120] V. H. Kshirsagar, A. M. Kanthe, and D. Simunic, “Trust based detection and elimination of packet drop attack in the mobile ad-hoc networks,” *Wireless Personal Communications*, vol. 100, no. 2, pp. 311–320, 2018.
- [121] B. Wang, X. Chen, and W. Chang, “A light-weight trust-based qos routing algorithm for ad hoc networks,” *Pervasive and Mobile Computing*, vol. 13, pp. 164–180, 2014.
- [122] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, “Fairness in wireless networks: Issues, measures and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.
- [123] D. M. Carballo, P. Roscoe, and G. M. Feinman, “Cooperation and collective action in the cultural evolution of complex societies,” *Journal of Archaeological Method and Theory*, vol. 21, no. 1, pp. 98–133, 2014.
- [124] E. Sørensen and J. Torfing, *Theories of democratic network governance*. Springer, 2016.
- [125] C. Yu, M. Zhang, F. Ren, and X. Luo, “Emergence of social norms through collective learning in networked agent societies,” in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 475–482.
- [126] A. Artikis and J. Pitt, “A formal model of open agent societies,” in *Proceedings of the Fifth International Conference on Autonomous Agents*, ser. AGENTS ’01. New York, NY, USA: ACM, 2001, pp. 192–193. [Online]. Available: <http://doi.acm.org/10.1145/375735.376108>
- [127] J. Pitt, J. Schaumeier, and A. Artikis, “Axiomatization of socio-economic principles for self-organizing institutions: Concepts, experiments and challenges,” *ACM Trans. Auton. Adapt. Syst.*, vol. 7, no. 4, pp. 39:1–39:39, Dec. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2382570.2382575>
- [128] A. Artikis, L. Kamara, J. Pitt, and M. Sergot, “A protocol for resource sharing in norm-governed ad hoc networks,” in *Declarative Agent Languages and Technologies II*, J. Leite, A. Omicini, P. Torroni, and p. Yolum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 221–238.
- [129] P. Faratin, C. Sierra, and N. R. Jennings, “Negotiation decision functions for autonomous agents,” *Robotics and Autonomous Systems*, vol. 24, no. 3, pp. 159–182, 1998.
- [130] B. An, V. Lesser, and K. M. Sim, “Strategic agents for multi-resource negotiation,” *Autonomous Agents and Multi-Agent Systems*, vol. 23, no. 1, pp. 114–153, 2011.

- [131] T. Baarslag, K. Fujita, E. H. Gerding, K. Hindriks, T. Ito, N. R. Jennings, C. Jonker, S. Kraus, R. Lin, V. Robu *et al.*, “Evaluating practical negotiating agents: Results and analysis of the 2011 international competition,” *Artificial Intelligence*, vol. 198, pp. 73–103, 2013.
- [132] T. Baarslag and K. V. Hindriks, “Accepting optimally in automated negotiation with incomplete information,” in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 715–722.
- [133] P. Kollock, “Social dilemmas: The anatomy of cooperation,” *Annual review of sociology*, vol. 24, no. 1, pp. 183–214, 1998.
- [134] M. M. Waldrop, *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, 1993.
- [135] P. E. Petruzzi, D. Busquets, and J. Pitt, “A generic social capital framework for optimising self-organised collective action,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2015 IEEE 9th International Conference on*. IEEE, 2015, pp. 21–30.
- [136] L. Buttyán and J.-P. Hubaux, “Enforcing service availability in mobile ad-hoc wans,” in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*. IEEE, 2000, pp. 87–96.
- [137] G. F. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, “Cooperation enforcement schemes for manets: A survey,” *Wireless Communications and Mobile Computing*, vol. 6, no. 3, pp. 319–332, 2006.
- [138] M. A. Nowak, A. Sasaki, C. Taylor, and D. Fudenberg, “Emergence of cooperation and evolutionary stability in finite populations,” *Nature*, vol. 428, no. 6983, pp. 646–650, 2004.
- [139] K. Sigmund, *The calculus of selfishness*. Princeton University Press, 2010.
- [140] J. Olejarz, W. Ghang, and M. A. Nowak, “Indirect reciprocity with optional interactions and private information,” *Games*, vol. 6, no. 4, pp. 438–457, 2015.
- [141] T. Kohler, J.-P. Steghöfer, D. Busquets, and J. Pitt, “The value of fairness: trade-offs in repeated dynamic resource allocation,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2014 IEEE Eighth International Conference on*. IEEE, 2014, pp. 1–10.
- [142] M. Olson, *The logic of collective action*. Harvard University Press, 2009, vol. 124.

- [143] E. Ostrom, *Governing the commons: The evolution of institutions for collective action.* Cambridge university press, 1990.
- [144] C. Hilbe, S. Sims, K. Chatterjee, and M. A. Nowak, “Evolution of cooperation in stochastic games,” *Nature*, vol. 559, no. 7713, p. 246, 2018.
- [145] S. Gächter, “Conditional cooperation: Behavioral regularities from the lab and the field and their policy implications,” CeDEX Discussion Paper, The University of Nottingham, Tech. Rep., 2006.
- [146] J. Pitt and J. Schaumeier, “Provision and appropriation of common-pool resources without full disclosure,” in *International Conference on Principles and Practice of Multi-Agent Systems.* Springer, 2012, pp. 199–213.
- [147] J. Konow, “Which is the fairest one of all? a positive analysis of justice theories,” *Journal of economic literature*, vol. 41, no. 4, pp. 1188–1239, 2003.
- [148] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations.* Cambridge University Press, 2008.
- [149] R. L. Haupt and S. Ellen Haupt, “Practical genetic algorithms,” 2004.
- [150] S. M. Pedersen and K. M. Lind, *Precision Agriculture: Technology and Economic Perspectives.* Springer, 2017.
- [151] J. Lindblom, C. Lundström, M. Ljung, and A. Jonsson, “Promoting sustainable intensification in precision agriculture: review of decision support systems development and strategies,” *Precision Agriculture*, vol. 18, no. 3, pp. 309–331, 2017.
- [152] V. Dignum, “Social agents: bridging simulation and engineering,” *Communications of the ACM*, vol. 60, no. 11, pp. 32–34, 2017.
- [153] S. F. Railsback and V. Grimm, *Agent-based and individual-based modeling: a practical introduction.* Princeton university press, 2012.
- [154] M. Wooldridge, *An introduction to multiagent systems.* John Wiley & Sons, 2009.
- [155] Open-Mesh. (2011) Batman-adv routing protocol. [Online]. Available: <https://www.open-mesh.org/projects/batman-adv/wiki>