sustainability

Article

The Development of a Service System for Facilitating Food Resource Allocation and Service Exchange

Viet-Cuong Trieu and Fu-Ren Lin *

Institute of Service Science, National Tsing Hua University, Hsinchu 30013, Taiwan
* Correspondence: frlin@iss.nthu.edu.tw

Abstract: To address the current limitation of Alternative Food Networks (AFNs) in tackling urban–rural sustainability issues, this study proposes a Cooperative Human-Agent Service System (CHASS) by leveraging the decentralized communication and coordination capability of a multi-agent system. The unique feature of CHASS is the collaboration between humans and agents for real-world deployment. From the perspective of Service-Dominant Logic (S-DL), value is co-created by involved actors through service exchange; that is, one actor's service exchanges for other one's service. With S-DL, technology is treated as an essential actant for resource integration, and the customer is a value co-creator. In this study, we propose a two-phase top trading cycle (TTC) negotiation mechanism to facilitate food resource allocation and service exchange. An agent-based model is developed to simulate the real-world environment and is integrated with CHASS to form a multi-agent simulation for system evaluation. In addition, to generalize the research outcomes, we use regression analysis to clarify the interaction mechanism between the algorithms applied by the platform and human decisions under the moderation of environmental factors. The results show the effectiveness of TTC-Negotiation mechanism to support resource allocation between customers and providers on CHASS. It shows the applicability of CHASS to the cooperative AFNs model.

Keywords: service system design; value co-creation; food resource allocation; cooperative food systems; alternative food networks; multi-agent system; multi-agent simulation; agent-based model; Service-Dominant Logic (S-DL)

check for updates

Citation: Trieu, V.-C.; Lin, F.-R. The Development of a Service System for Facilitating Food Resource Allocation and Service Exchange. *Sustainability* **2022**, *14*, 11987. https://doi.org/10.3390/su141911987

Academic Editors: Rashid Mehmood, Tan Yigitcanlar and Juan M. Corchado

Received: 30 July 2022
Accepted: 17 September 2022
Published: 22 September 2022

1. Introduction

The Conventional Food System (CFS) has become industrialized and globalized, significantly impacting food and nutrition security. In reaction to the industrialization and globalization of the CFS, the Alternative Food Networks (AFNs) initiative emerged in the 1990s. AFNs seek to diversify and transform modern food provision by connecting ethical producers and consumers in more local, direct ways [1] and creating a trusting relationship in the food network. Although the AFN initiative aims to contribute to sustainability, there are also many criticisms regarding its unsustainability [2]. The most critical criticism concerns economic sustainability. For example, while removing intermediaries in the supply chain promises to redistribute more value to producers, selling directly to customers on a small scale may require more resources from producers, and the net benefit may not live up to the theory [3]. In addition, "value-added" products with a higher price may benefit the producer but be out of the reach of lower-income customers [4]. To increase economic efficiency, some AFNs have evolved towards forming an intermediary organization and using centralized digital platforms for serving more customers in busy modern life. However, such models will somewhat lose an essential feature of AFNs, which is the establishment of "strong relationship" between customers and producers.

Today's digital services are highly centralized. The dominant client–server architecture favors the centralized ownership of servers [5], causing a lack of interoperability, which leads to centralizing economic control and data. Furthermore, a centralized IT platform is probably not technically sustainable. With the recent rise of decentralized Web 3.0 [6], it is evident that alternative, decentralized systems can be technologically and economically sustained. From a Service-Dominant Logic (S-DL) perspective, value is co-created by involved actors through service exchange; that is, one actor's service exchanges for other one's service [7,8]. Therefore, a decentralized information system can become an essential digital service for resource integration and value co-creation, on which consumers can play a role in value co-creation through information sharing and collaborative decision-making.

The objective of the study is to develop a decentralized service system as an innovative IT resource to help overcome the limitations of the AFN models and achieve technical sustainability. To achieve this goal, in this study, we propose a decentralized Cooperative Human-Agent Service System (CHASS), which supports service exchange in which the human actor has the agency to control and is assisted by an intelligent agent. CHASS is designed to achieve two goals: *(1) to improve food resource allocation performance through an multiagent-enabled negotiation mechanism* and *(2) to strengthen human relationships through service exchange*. Through designing, building, and evaluating the CHASS, this study will answer the research question "*How does a multiagent-enabled negotiation mechanism influence the service exchange under the moderation of different supply-demand patterns and customer's collaborative decision-making*."

This study was conducted in the context of Taiwan, which is affected by urbanization and rural decline. Since the late 2000s, the Taiwan government has been promoting many "rural revitalization" programs in response to this problem. However, although the impact of these government funding projects has claimed some success, some projects are no longer effective after the government funding ends. Moreover, taking a lesson from severe rural decline and urban–rural imbalance in Japan, using only financial resources seems insufficient to prevent rural decline and promote urban and rural linkage. Developing AFNs is an approach to strengthen the linkage between urban and surrounding rural areas and contribute to rural sustainability. This motivates this research to develop CHASS to contribute to the creation of an economically sustainable AFN model, which promises to create more value for rural areas and strengthen urban–rural linkages in Taiwan. Besides the contribution to the design artifact, design knowledge, and evaluation methods, this study has developed CHASS as a practical service system that can be used for performing actor-to-actor communication and coordination to achieve the sustainability of the service ecosystems. With the spirit of "engaged scholarship", CHASS will be further developed to contribute to sustainability in various domains, such as cooperatives, the circular economy, and community of practice in knowledge sharing and collaboration.

2. Literature Review

2.1. Alternative Food Networks (AFNs)

The term "Alternative Food Networks" (AFNs) has emerged both in practice and in academic works since the 1990s as a reaction against the standardization, globalization, and unethical nature of the "conventional food system" [1]. AFNs promote food consumption closer to the producer and change consumer culture and behavior. The three core characteristics of AFNs are: (1) the requirement for products and production, (2) the closer distance between producers and consumers, and (3) new forms of food market governance [2]. The products in AFNs are often characterized as fresh, natural, organic, quality, and "slow", while the production is described as environmentally benign using traditional production methods [9]. The distance between producers and consumers is measured in three dimensions: physical distance, supply chain distance (the number of intermediaries), and informational distance (the availability of information in the network). AFNs also generate new forms of food markets such as Community Supported Agriculture (CSA), farmer market, and consumer and producer cooperatives.

2.1.1. The Promise of AFNs on Sustainability

The AFN initiative promises to contribute to all economic, environmental, and social sustainability aspects. In the *economic aspect*, AFNs improve producers' income by adding value through differentiated production methods [10], and social embeddedness enables the customer to accept a higher price [11]. Arrangements such as CSA are built on sharing the economic risk between producers and consumers [12], while the key ideas of another arrangement (e.g., producer cooperatives) create more negotiating power and resources for producers, thus better market possibilities and income. The reduction of supply chain distance allows a higher share of value to be captured by the producer [13]. Selling directly through the market is an excellent way for some small producers to get to the market [14]. Furthermore, the reduced physical distance in AFNs means more money is spent on local food and contributes more to the local economy.

In the *environmental aspect*, environmentally benign production choices can positively contribute to sustainability [15]. The reduced physical distance in AFNs reduces food transportation distances, equaling less fuel use and emissions to the air [16]. In the *social aspect*, food quality and production contribute to consumer health. "Natural" foods are believed to be healthier than highly processed foods. The reduced physical distance keeps food fresher and retains more nutrients [17]. Environmentally benign production methods can contribute to producer and consumer health and safety. Organic farming, for example, restricts the use of chemicals, thus addressing product and producer safety, health impacts, and biodiversity. Local and diverse production is considered critical for food security [16,18,19] and positively affects food culture. Traditional production methods can contribute to the preservation of traditional food cultures and diversity [16].

AFNs may also create indirect sustainability impacts related to learning and participation. For example, the reduced informational distance contributes to increasing participants' learning about and awareness of sustainability-related issues in the food system. Increased learning and awareness, in turn, are believed to lead to more sustainable practices [20,21]. In addition, these indirect impacts can reinforce participant choices about preferred production methods, the form and length of food supply chains, and governance arrangements [2].

2.1.2. Limitations of AFNs

Although the AFN initiative contributes much to sustainability, it also faces criticism. In terms of *economic aspect*, the reduced value chain distance raises the question of how value is being redistributed and what the net benefit to producers is. Many studies suggest that direct selling to consumers may require more resources, time, and energy from producers, and the net benefit may not live up to the theory [3]. For example, from an empirical study on the farmer's market, James [22] raised the question: Have farmers received enough income from farmers' markets? If there is a lack of customers, farmers may suffer losses, including financial loss and an accompanying loss of working time on the farm. Therefore, the farmers could not economically sustain attendance at the market, and farmers' markets could not provide a viable alternative for small-scale producers.

In terms of the *environment aspect*, reduced "food miles" may not be as significant as reducing transport-related emissions, and small-scale food distribution may be inefficient in the means of transport [23]. Transportation also generally causes only a tiny part of the life-cycle greenhouse gas emissions of food [24,25]. In terms of *social aspect*, fresher and more nutritious food due to shorter physical distance has been challenged by considering the time, not just distance in transport [17]. "Value-added" products with a higher price may benefit the producer but be out of the reach of lower-income consumers [4].

While AFNs represent a broad group of alternative agricultural relationships and practices, an important goal is to establish a "strong relationship" and trust between consumers and producers. However, the practical AFN forms nowadays do not establish "strong relationship" as expected due to the lack of regular interaction between producer and consumer. In general, most AFN models in practice can achieve the goal of reducing the physical distance and supply chain, but there are still many limitations in reducing the information distance.

2.1.3. Community Supported Agriculture (CSA) Model

The CSA is probably the best model to achieve the "strong relationship" goal among the many forms of AFNs. Initially, the CSA model consists of a community of individuals who pledge support to a farm operation where the growers and consumers provide mutual support, sharing the risks and benefits of food production [26]. The core value of the traditional CSA is the "support" and "cooperation" of the community. However, over time, the definition of CSA has changed significantly, and most CSA business models today operate in a very different way from the early CSA model. CSA models have shifted from emphasizing the farmer (community supports farmers) to focusing on the customer (how CSA works better for customers) [27].

Although the modern CSA models are also very diverse and different, product quality commitment and a close distance between customers and producers are standard features. However, the modern CSA models have gradually lost the original feature: community cooperation. That is a shift from the long-term commitment (e.g., annual subscription) to a shorter commitment (e.g., weekly subscription) and the emergence of the no-commitment model, the "pay-as-you-go" CSA [28]. The positive in the evolution of CSA models is that it has reached more customers, bringing more value to customers and producers. However, the "strong relationship" goal between the consumer and the producer has been traded off. In the modern CSA models, customers view the CSA as an organization from which they can buy products rather than cooperate with the producer.

2.1.4. Alternative Food Networks and Urban–Rural Linkage

In the ecosystem view based on "rural–urban linkage", food provision is an essential service in the relationship between urban and rural areas [29]. Therefore, strengthening food service is considered the most effective solution among existing solutions to strengthen rural–urban linkage. Over the last century, the food supply chain has become more industrialized and globalized, and these significant changes directly impact the urban–rural relationship. As a result, urban and rural areas have become less interdependent on food service systems. To react against the standardization, globalization, and unethical nature of the industrial food service system, "Alternative Food Networks" (AFNs) emerged. Therefore, developing AFNs is an approach to strengthen the linkage between urban and surrounding rural areas [30]. Furthermore, AFNs also play a crucial role in food security for urban and rural populations.

2.2. Service-Dominant Logic (S-DL)

The S-DL is a research stream that has emanated over the last 20 years from a concern about the traditional understanding of service(s). The S-DL identifies service –the process of using one's resources for the benefit of another actor—rather than goods as the fundamental basis of economic (and social) exchange [7]. In S-DL, the role of resources is central to the process of value creation, which occurs 'when a potential resource is turned into a specific benefit [8]. Moreover, the S-DL emphasizes the role of operant (intangible) resources that are capable of acting on operand (tangible) resources and even other operant resources to create value [8]. Technology is both an operand resource (e.g., system platform) and an operant resource, a combination of practices, processes, and symbols that fulfill the human purpose [31].

This study approaches the relationship between customers and producers in AFNs under the S-DL perspective: customers as a co-production and co-creation value rather than the receiver of value [32]. In S-DL, all participants with different roles (e.g., producers, customers) are viewed as actors in networks of other actors (A2A network) and co-creating value through resource integration and service provision [33]. The S-DL emphasizes information technology, which is an essential resource in the process of resource integration and value co-creation, through two concepts: resource liquefaction and resource density. Resource liquefaction refers to the possibility of decoupling information from its physical form to share it with others and become more valuable. The resource density is the level at which resources are quickly mobilized for a time/space/actor. Maximum density occurs when the best combination of resources is mobilized for a particular situation [34].

2.3. Food Preferences, Food Choice, and Demand Flexibility

Consumer preferences are the subjective (individual) tastes of various bundles of goods. It can be considered common knowledge that people have different food preferences. Some people like a variety of foods, while others may be picky eaters. Biological, psychological, and sociocultural factors influence food preferences [35]. The process of connecting food preferences and choices is not straightforward. According to Wadolowska, et al. [36], food preferences interact with different food choice factors (e.g., advertising, functional value, health, and price) and consumer socio-demographics (e.g., age, economic condition, education, gender). It might seem that we often make food choices based on intuitive thinking that are not consciously monitored, resulting in effortless and fast decisions [37].

Demand flexibility refers to the consumer demand that can be reduced, increased, or shifted to substitute products within a specific time. Demand flexibility is an essential concept in the energy field in the context of climate change because renewable energy is only generated for a particular time [38]. Under the S-DL perspective, the consumer can play a role as a value co-creator if they are flexible enough to adjust their consumption for the common good between themselves and service providers. "Demand flexibility" in this study refers to customers' flexibility to switch from a product to another substitute product in AFNs. The "food choice" factors (e.g., advertising or health) mentioned above are similar for different products within the same AFNs. Therefore, food preferences are the main factor affecting product switching acceptance. Demand flexibility can be viewed as a resource in a food system because it helps reduce food storage and spoilage costs.

2.4. Priority-Based Resource Allocation Algorithm

The priority-based resource allocation is a commonly observed problem in real-life. In this problem, the resources are allocated based on the participants' preferences and priorities. This section will review the top trading cycles (TTC), initially proposed by Shapley and Scarf [39], a popular algorithm to deal with the priority-based allocation problem. TTC is a unique core matching, Pareto efficiency, strategy-proof, and individually rational algorithm [40].

In the TTC algorithm, each person or object is presented by a node in a directed graph (Note: The graph in this context is made up of vertices (nodes), which are connected by edges (links) in graph theory). A person node points to its most preferred object node, and an object node points to its highest-priority person node, as shown in Figure 1. The allocation process starts with determining cycles in the graph. There is at least one cycle in the graph. Then, the nodes belonging to a cycle are allocated and removed from the graph. Finally, the same procedure is applied to reduce the market participant (nodes in the graph). If the object has multi-units, a counter is assigned to each object to denote available units at each allocation step. Many studies have developed TTC-based mechanisms to solve practical problems such as the school choice problem [41], kidney allocation [42], house allocation [43], landing slots assignment among flights [44], and tuition and worker exchanges [45].

2.5. Agent Systems

There is no strict definition of "agent" due to the rapid growth of the diversity and functionality of agents in different disciplines. In general, an agent can be defined as an entity with two important capabilities: (1) the capability to act with a certain degree of autonomy and (2) the capability to interact with other agents to engage in analogs of the kind of human social activity: cooperation, coordination, negotiation, and the like [46].

This section will describe a Multi-Agent System (MAS), an Agent-Based Model (ABM), multi-agent simulation, and agent negotiation.

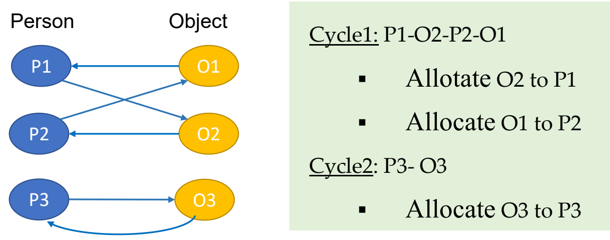
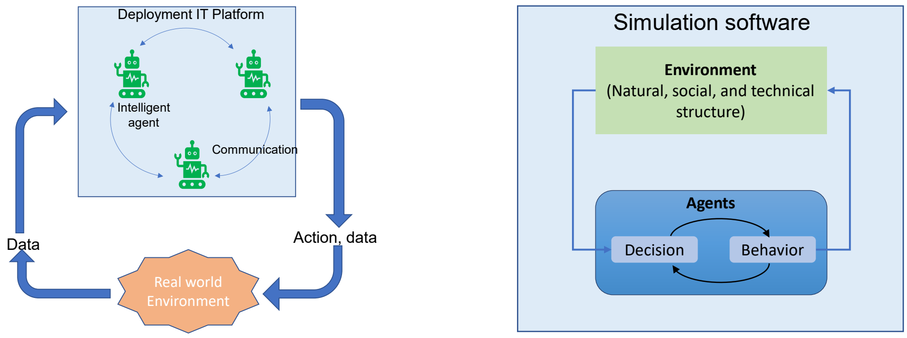


Figure 1. A simple illustration of the TTC algorithm.

2.5.1. Multi-Agent System (MAS) and Agent-Based Model (ABM)

A Multi-Agent System (MAS), as shown in Figure 2a, is a self-organized system composed of multiple agents interacting to reach goals that are difficult or impossible for an individual agent to achieve. MAS is mainly studied in the field of computer engineering to solve practical or engineering problems, and as a distributed system, agents in MAS can deploy and operate in different physical devices. A significant milestone in developing multi-agent systems is the "Distributed Artificial Intelligence" workshop, held at MIT in June 1980 [47]. Then this field gained widespread recognition in the mid-1990s and has grown enormously since then.



(**a**) (**b**)

Figure 2. (**a**) A Multi-agent system, (**b**) an Agent-based model.

Although there is considerable overlap with the MAS, the goal of an ABM is different. The ABM (Figure 2b) is a computational model for simulating the actions and interactions of autonomous agents to understand a system's behavior and what governs its outcomes. ABM helps to understand collective behavior's effects by representing the rules governing agent decisions and the influence of these decisions in a real-world environment. ABM components, including agents and environment (natural, social, and technical structure), are simulated by computer software [48]. The outcome of the agent decision-making process can be directly affected by past individual behavior and the behavior of other agents. In addition, the agent's decision-making can be influenced by the environment. The ABM is used in many scientific domains, such as biology, medicine and health, ecology, and social science [49]. In summary, the ABM aims to explore insight into the agent and system behavior through simulation, while MAS is an IT system that was developed to solve practical problems.

2.5.2. Multi-Agent Simulation

The simulation is considered a computational tool to achieve two major goals: understanding a real system and developing a real operational system. According to the survey of Michel et al. [50], a large number of research works and software applications belong to the intersection between MAS and simulation, which is separated into two approaches: (1) *Simulation for MAS* and (2) *MAS for Simulation*.

The first approach, *Simulation for MAS*, refers to projects wherein computer simulation is used to design, experiment, study, and run a MAS architecture. Simulation is generally recognized as one of the best design support technologies when designing a complex, dynamic, and stochastic system. This approach is similar to the software-in-the-loop approach in the software development field. The simulation method allows developers to experiment with MAS in a controlled and cost-effective manner, using simulation-run contexts instead of the actual run. In addition, the developers can gain extensive empirical experience with the essential issues in designing distributed problem-solving systems through the simulation process. Following this approach, the self-organizing MAS will be developed independently of the simulation tools.

The second approach, *MAS for Simulation*, is related to simulation experiments that use MAS as a modeling paradigm to build artificial laboratories. This approach is well-known as ABM. An agent-based simulation is a bottom-up approach wherein humans could explicitly define a simulated agent's decision processes at the micro-level in a multi-agent simulation model. Structures emerge at the macro level due to the agents' actions and interactions with other agents and the environment. The ABM helps us to understand how real systems' dynamics arises from individuals' characteristics and environments. Furthermore, it allows modeling a heterogeneous population in which each actor can have individual motivations and incentives while at the same time representing groups and group interactions.

2.5.3. Agents Negotiation

In MAS, the negotiation mechanism allows agents to negotiate to reach an optimal solution despite this lack of the complete knowledge of the environment [51]. The primary purpose of the negotiation algorithm is to find an optimal solution to a negotiation problem. The negotiation process is triggered by an autonomous coordinator agent, as shown in Figure 3. The coordinator agent initiates the negotiation event and manages the whole negotiation process. Besides the coordinator agent, there are several or more collaborating agents. This collaborating agent is named as the assistant agent in this study because it assists human users. The assistant agent is a semi-autonomous agent who can make a decision on behalf of human users. Assistant agents participate in the negotiation process to maximize utility based on their principals' preferences. Meanwhile, the coordinator agent will operate the negotiation process to maximize resource allocation. The negotiation process ends when the maximum resource allocation is reached.

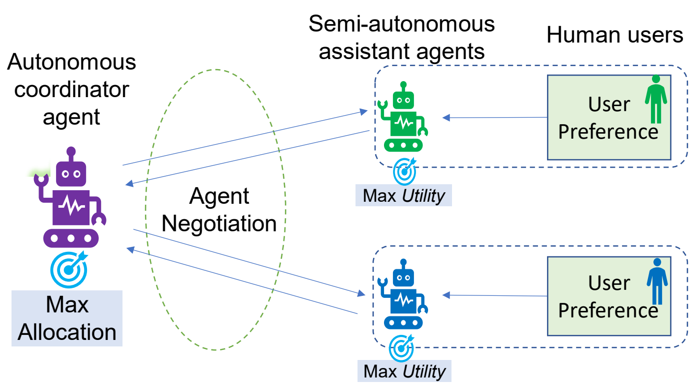


Figure 3. The MAS negotiation mechanism.

2.6. The Next MAS Generation for Human–Agent Societies

MAS technology allows the development of autonomous agents that are naturally designed to communicate with each other. This communication enables complex interactions, from which higher-level social activities such as cooperation or collaboration may emerge [52]. Although MAS has been studied for many years, the progression in real applications in which agents can interact and assist humans in daily life has not yet reached the expected levels. Integrating humans and intelligent agents in the same system will be one key challenge in the next generation of MAS. Therefore, the next generation of MAS technology needs to support the development of applications wherein agents and humans can jointly provide services to other humans or agents in a "human–agent society." [53].

Human–agent societies can be developed based on the next generation of MAS that supports a natural interface for interaction between humans and agents. At the agent level, humans are integrated into the system in such a way that they appear as agents to other agents. While at the human level, the human can interact with agents using natural language. In addition, transparently communicating humans with agents will be vital in developing fully open systems wherein entities (humans, agents, or third-party elements) can dynamically enter or exit the system transparently [52]. This open system feature has traditionally been a challenge to be realized for real-world applications. However, it has become possible based on the development of natural language processing technology that allows human–agent interaction through natural language combined with the availability of an appropriate communication protocol and infrastructure.

2.7. Previous Related Studies:

This section will present three recent research directions related to this research, including the research direction of demand flexibility and MAS in the energy field, the human-in-the-loop in MAS, and the service value co-creation digital platform for sustainability.

2.7.1. Demand Flexibility and MAS for Resource Allocation Optimization

In the energy field, demand flexibility is the capacity to shift electricity consumption over time or to switch to renewable energy such as solar power and wind power locally to obtain economic benefits and achieve sustainable production and consumption goals. The research stream on MAS to solve the energy optimization problem has also been interested by many researchers and practitioners in recent years [54]. Under the MAS approach, each family will install an intelligent device acting as an autonomous agent to communicate, coordinate, and cooperate with other agents to reach demand flexibility [55]. In this study, we bring the concept of "demand flexibility" from the energy sector to the food sector and develop a MAS to achieve food-demand flexibility.

2.7.2. Human-in-the-Loop in MAS

The difference between MAS in the energy sector and the food sector is the involvement of a human actor. Each individual has a different food preference, which also changes over time. Therefore, the MAS in the food sector must be designed to allow humans to collaborate with the agent in the food choice decision-making process. Recent achievements in natural language processing have allowed the design of human–machine interaction systems through natural language. Some research directions related to the human-in-the-hoop stream can be mentioned, such as improving collaboration between humans and machines [56] and the interaction between humans and agents in the context of a smart city [57].

2.7.3. Service Value Co-Creation in a Digital Platform for Sustainability

A digital platform can play the role of an operand resource that can help hold together diverse actors and enable collaboration in the ecosystem. From the S-DL perspective, the service platform comprises tangible and intangible components that facilitate the interaction of actors and resources, leveraging resource liquefaction and enhancing resource density [58]. Two types of service value co-creation contribute to the sustainability mentioned in recent studies: second-hand trading platforms that enable sustainable consumption [59] and CSA digital platforms that support collaboration between members in the CSA model in

AFNs [60]. This study develops MAS to work as a decentralized service value co-creation platform for sustainability.

3. Research Method

This study adopted the Design Science Research (DSR) method in information systems [61]. The DSR method focuses on the "design" process for building and evaluating the IT artifacts to solve problems. In design science, knowledge of the problem domain and its solution is achieved through the building and application of the designed artifact. Following the guidelines for design science in information systems research proposed by Hevner, March, Park and Ram [61], this study was conducted in four stages: (1) Problem formulation, (2) Design and building, (3) Design and evaluation, and (4) Reflection and formalization of learning.

3.1. Problem Formulation

The research problem is formed based on the author's experience, observation, and interview when participating in "university-community" projects in Taiwan, combined with a careful review of the limitations of the "regional revitalization" projects in Japan and Taiwan and a thorough review of AFN literature and other relevant studies. This stage follows the "problem relevance" guideline of the DSR method: the objective of design-science research is to develop technology-based solutions to important and relevant community problems [61].

3.2. Design and Building

This stage follows the guideline "design as an artifact" of the DSR method: design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. We developed a MAS to solve the community problem.

In the design step, we combine the Prometheus method [62] and the Behavior-Oriented Design (BOD) method [63] to develop intelligent agents. The Prometheus method consists of three phases: (1) system specification, (2) architectural design, and (3) detailed design. We follow the first two phases of the Prometheus method: system specification focuses on identifying the basic functionalities of the system and architectural design to determine which agents the system will contain and how they will interact. We then design the agent in detail based on the BOD approach. The BOD is a development method for designing a complex agent, an agent that can function naturally on its own [63]. The BOD approach helps decompose the complex agent into simple behavior, so we can rapidly develop system prototyping using MAS middleware.

Based on the agent behavior design, we use SPADE [52], a middleware for multi-agent systems, to build the platform. SPADE supports five behavior types: cyclic, one-shot, periodic, time-out, and finite state machine, which can support the development of rapid system prototyping.

3.3. Design and Evaluation

This stage follows the "design evaluation" and "design as a search process" guidelines of the DSR method. The "design" and "evaluation" activities are integrated into a cyclic process to discover an effective solution. We use the multi-agent simulation method to conduct an intervention and evaluation. The intervention can lead to both "intended" and "unintended" outcomes. These "unintended" outcomes are essential factors for reshaping the design process.

We built a simulation by integrating the ABM with MAS. First, we designed and built an ABM using the Mesa framework [64] to model the real-world environment. Next, the ABM is set up to generate data similar to the data created by human users during interaction with the designed MAS. Then, we perform the intervention by setting different simulation parameters (environment, resources, rules, and decision model of simulated agent) in ABM to generate input data for the MAS, and then the output data of MAS is collected and analyzed. For each simulation, the results are preliminarily evaluated and stored for later analysis.

3.4. Reflection and Formalization of Learning

Besides contributing to an innovative design artifact, this study contributes to service system design knowledge and evaluation methods. We synthesize the design knowledge through the process of designing and building the service system to solve a community problem. Furthermore, through intentional intervention and evaluation using multi-agent simulation, we can better understand under what circumstances the negotiation process will perform better. From then, a new requirement for the service system emerges. For example, by understanding the interaction mechanism of demand flexibility and the negotiation process on resource allocation performance, we know in which situation the system needs to provide incentives to encourage users to accept negotiation. Since then, the system requirements can be revised to develop the platform's next version.

In addition, we use regression analysis and a t-test to answer the research question: *How does a multiagent-enabled negotiation mechanism influence the service exchange?* The regression analysis results will shed light on the influencing mechanism of negotiation on the "resource allocation" and "order fulfillment" performance under the moderation of supply/demand patterns and the degree of customer demand flexibility. This step creates a clear understanding of the designed service system and generalizes the research outcomes.

4. Cooperative Human–Agent Service System (CHASS)

4.1. System Design Principle

A Cooperative Human–Agent Service System (CHASS) was designed based on three design principles: (1) efficient resource allocation, (2) strengthening actors' relationships in the A2A network, and (3) ensuring technical and economic sustainability.

The first and second design principles are the functional requirements to overcome the limitations of the current AFN forms. From the S-DL perspective, CHASS plays a role as a digital resource for integrating other resources through allocation, collaboration, and negotiation mechanisms. In addition, CHASS is designed to strengthen human relationships by giving human autonomy to select partners for service exchange. This is necessary to create a strong relationship between customers and producers and contribute to urban–rural sustainability.

The third design principle is a requirement to ensure the service system can maintain its capacity to sustain itself. A service system aiming to achieve sustainability goals needs to achieve economic and technical sustainability. To be technologically sustained, the service platform must be completely open, decentralized, and evolved based on community resources. In addition, the system platform must generate economic value higher than costs in terms of economic sustainability. Decentralized systems can contribute to economic sustainability since collectively supporting and contributing to the system operation could share the centralized operation and maintenance costs. This principle can be realized based on the following two requirements: (1) *distributed environment*, agents can operate anywhere on the public Internet and are flexibly deployable in differently resourced devices; (2) *decentralized control*, agents and humans can directly communicate for service exchange without a centralized authority.

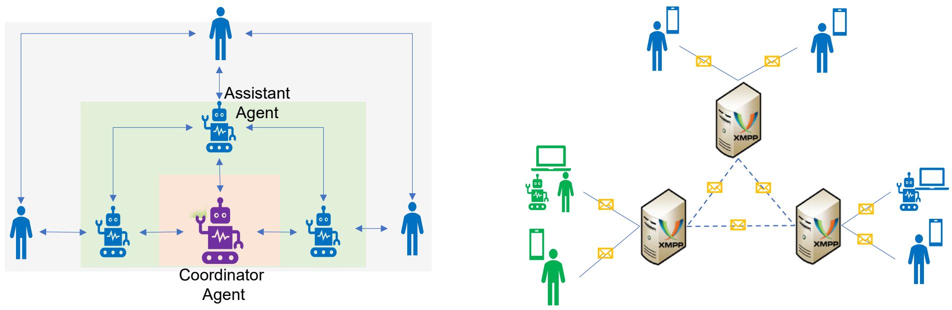
4.2. System Architecture

4.2.1. Distributed Environment

To achieve the technical sustainability goal, CHASS is designed to be an open system deployed in a distributed environment where agents can be deployed on any device with an Internet connection. A distributed system will bring the advantages of data privacy, user autonomy, system scalability, and stability. However, the economic benefits of a distributed system are generally unclear and controversial. A distributed system has no centralized operating and maintenance costs but generates other hard-to-estimate costs:

agent computation power and communication costs. The more agents on the network, the more communication resources will be wasted on the allocation and negotiation process.

In this study, CHASS is designed based on a hybrid centralized–distributed architecture [65] for archiving economic and technical sustainability principles, as shown in (Figure 4a). This architecture has two types of agents: coordinator agent and assistant agent. The role of the coordinator agent is to coordinate the resource allocation and negotiation process, thereby minimizing the dense communication in a network. The assistant agent works as a personal virtual assistant carrying out some set of operations on behalf of humans with some degree of independence to generate value for its owner. Humans can communicate with their virtual assistants through a mobile app using natural language. From a service view, this architecture is designed based on the coordinative cooperation approach [66]: the coordinator agent invites the assistant agent to join in cooperation with other agents to achieve a common goal.



(**a**) (**b**)

Figure 4. (**a**) The system architecture of CHASS and (**b**) decentralized communication in CHASS.

4.2.2. Decentralized Communication

In CHASS, the agent's communication is done over the XMPP (eXtensible Messaging and Presence Protocol), an open protocol for instant messaging that has been widely used in the industry (e.g., WhatsApp, Google Talk). In addition, XMPP uses a presence notification mechanism [67] that any entity may enact by providing a list of other entities as contacts and requesting to be notified when any contact changes their state.

XMPP supports a decentralized server architecture (Figure 4b) that supports agents and humans in directly communicating without a centralized authority. In addition, this decentralized architecture allows agents to be independent of the device's IP address where they are running. This feature makes a difference compared with many other MAS that rely on IP addresses to send and receive messages. In addition, agents are identified by the XMPP server where they are registered, not by the device where they are running; therefore, agents can migrate from one device to another transparently. Security is another crucial issue for MAS platforms deployed in the real world. XMPP protocol also provides security at different levels, such as certificates and Transport Layer Security (TLS), to encrypt communications and sign messages to ensure they are sent and received by reliable endpoints.

In CHASS, there are three types of interactions: agent to agent, agent to human, and human to human. All interactions are done through the XMPP protocol. The communication between agents is designed based on the FIPA Agent Communication Language Specifications (FIPA ACL) [68]. Human–agent interaction is performed in natural language. Human-to-human communication is also important because CHASS aims to establish a strong human relationship. In CHASS, humans can directly communicate with each other via mobile chat App, similar to human–agent communication.

4.3. Service Flow in CHASS

The service flow for resource allocation and value exchange in CHASS is divided into four stages, as presented in Figure 5. Stage 1, *information acquisition*, begins after the assistant agents receive the supply and demand information from customers and providers. In this stage, the coordinator agent receives demand and preferences for products and providers from the customer's assistant agents and supply and customer priority from the provider's assistant agents. Next, during Stage 2, *allocation and negotiation*, the coordinator agent interacts with the customer assistant agent for food resource allocation and negotiation. When the allocation and negotiation process is complete, the coordinator agent sends the allocation results to the corresponding customer and provider agents (Stage 3—*allocation result informing*). Finally, in Stage 4, *service exchange*, customer assistant agents make order requests to the matched providers based on the allocation information received. The service exchange process (e.g., ordering, additional service requests, order confirmation, payment) is done directly between the providers and the customers. In this stage, customers and providers have autonomy and freedom for service exchange to benefit each other. The service flow in Figure 5 will be further clarified through the agent state–transition diagram for a coordinator agent and an assistant agent (Figures 6 and 7).

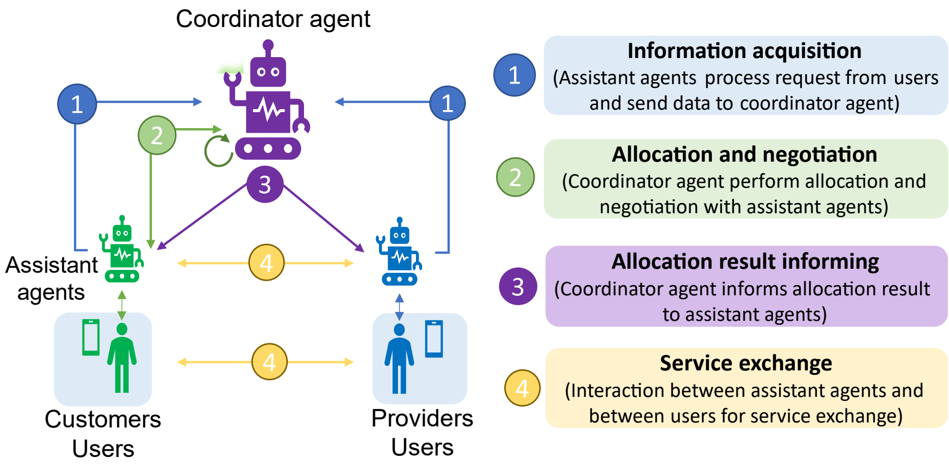


Figure 5. Resource allocation and service exchange flow in CHASS.

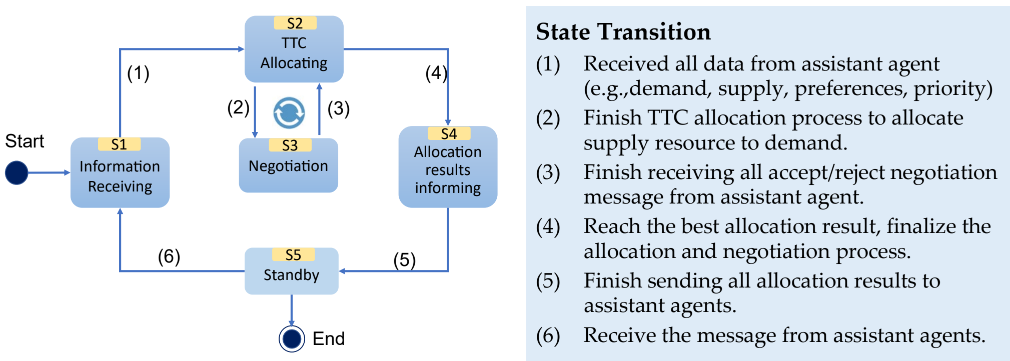


Figure 6. State–transition diagrams of a coordinator agent. (S1) Receiving demand/supply, (S2) running the TTC mechanism to allocate supply to demand, (S3) interacting with an assistant agent for negotiation, (S4) sending allocation results to all assistant agents, and (S5) standby.

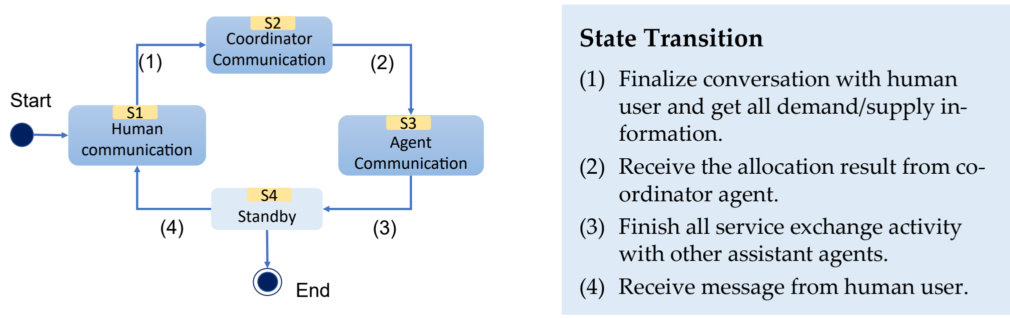
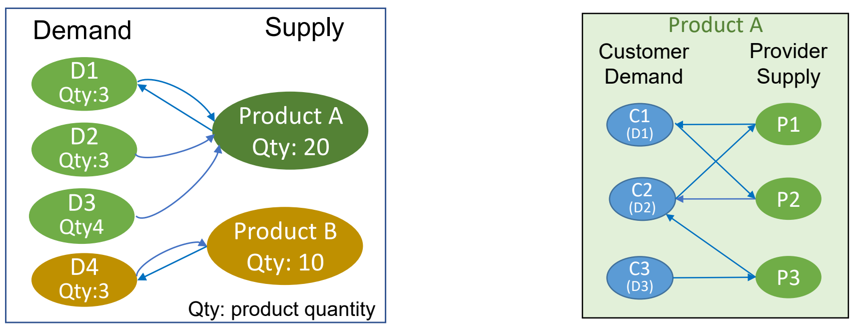


Figure 7. State–transition diagrams of an assistant agent. (S1) Communicating to humans to obtain demand or supply data, (S2) communicating to the coordinator agent to inform supply or demand and negotiation, (S3) interacting with other assistant agents for service exchange, and (S4) standby.

4.4. Two-Phase TTC-Negotiation Resource Allocation Mechanism

As mentioned in Section 4.3, in Stage 1, the coordinator agent receives supply and demand information from assistant agents, and then the supply information is grouped by product and aggregated in quantity. After completing information acquisition, the coordinator agent starts the allocation and negotiation process in Stage 2. This subsection describes in detail the proposed two-phase TTC-Negotiation resource allocation mechanism (Figure 8). In phase 1, the product resource is allocated to each demand based on the product preferences and demand priority. In phase 2, the allocated demand and supply are grouped by product, and the TTC algorithm is applied for each product group to bind each customer's demand to corresponding providers based on the priority both sides provide (customer and provider).



(**a**) (**b**)

Figure 8. Two-phase TTC-Negotiation mechanism: (**a**) Phase 1—Product allocation and negotiation; (**b**) Phase 2—Binding customer demands with providers.

Phase 1: Product allocation and negotiation

This phase aims to meet the "efficient resource allocation" design principles. The TTC-Negotiation mechanism is used to allocate product supply to each demand based on customer demand priority and the product preferences of customers. Demand priority can be determined based on demand time or using priority models, such as the RFM (Recency, Frequency, Monetary) model [69]. Figure 8a illustrates the product allocation process, which is executed through several rounds of operations specified as follows:

- Round 1: Each product supply is presented as a supply node and is assigned a counter equal to the product supply quantity. Each customer demand is represented as a demand node and is assigned a counter equal to the demand quantity. The demand nodes point to their first preference of product node, and each product node points to the highest-priority demand node. Each demand node in a cycle is allocated the product it points to and is removed from the graph. The counter of each product node in the cycle is reduced accordingly. If this counter value reaches zero, the product node will be removed. If there exists a demand node that does not point to any product node, it will also be removed.

- Round k, k ≥ 2: Round k uses the same allocation and removing procedure as Round 1. This iteration will stop if no demand nodes or product nodes are left.

After the first TTC allocation, the negotiation condition is checked (Figure 9). If all demand is fulfilled or all supply is allocated, it means no more demand or supply can be allocated, and the best allocation has been reached; then, the allocation process is finished without negotiation. On the other hand, if there is still unfulfilled demand and unallocated supply after the first TTC allocation, the coordinator agent will send a negotiation invitation to the target assistant agents. Target agents are agents who belong to one of two groups: (1) agents who have unfulfilled demands or (2) agents who have fulfilled demands with products in shortage. The goal of the negotiation process is to convince agents in group 2 to switch to products that are not yet allocated so that the demands of agents in group 1 can be fulfilled. Next, all assistant agents who accept negotiation will be formed into a negotiation group; then, the second TTC allocation is run for this group. This process repeats in several rounds until the best allocation results are reached.

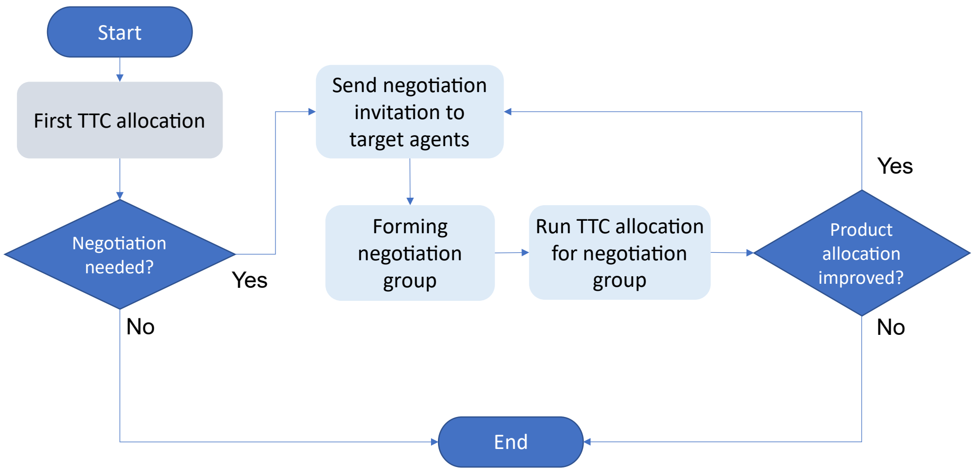


Figure 9. TTC allocation and negotiation flow. The negotiation process is carried out if there is unallocated supply and unfulfilled demand for substitute products.

Phase 2: Binding customer demand with provider

In phase 2, the customer demand and provider supply are bound based on priority provided by both sides (customers and providers). This phase aims to meet the design principles: (2) strengthening actors' relationships in the A2A network. The actor (customer or provider) has the autonomy to choose the exchange partners by providing a priority list. The assistant agent can be set up to compute other agents' priorities based on several criteria (e.g., product rating, trust level, etc.) on behalf of human users who set the priorities based on personal preference. If the assistant agent does not provide other actors priority, the coordinator agent will randomly generate the priority. Finally, the TTC mechanism is applied for each demand and supply group to bind each customer's demand to the corresponding providers based on the priority of both sides.

When is the negotiation needed?

The conditions for negotiation in Figure 9 are described more clearly through three scenarios presented in Table 1, in which X and Y are two interchangeable products. Scenario 1 is common in the CFS, where supply often exceeds demand and food waste occurs. Conversely, when events such as natural disasters occur, scenario 2 will occur. Scenario 3 is probably rare in CFS but possible in AFNs. In the cooperative AFN model, customers and producers are both members, and information is shared among members. Therefore, the cooperative will be able to adjust the total supply to meet the total demands of members to avoid wasting food or not providing enough for members. When scenario 3 occurs, the negotiation process will be activated.

Table 1. The three scenarios of supply and demand pattern.

Scenario		Product X Quantity	Product Y Quantity	Allocated and Fulfilled	Unallocated & Unfulfilled	Status	Negotiation
1	Demand	1	1	X: 1, Y: 1	-	All demand is fulfilled.	No
	Supply	2	2	X: 1, Y: 1	X: 1, Y: 1		
2	Demand	2	2	X: 1, Y: 1	X: 1, Y: 1	All supply is allocated.	No
	Supply	1	1	X: 1, Y: 1	-		
3	Demand	2	1	X: 1, Y: 1	X:1	unallocated supply and unfulfilled demand	Yes
	Supply	1	2	X: 1, Y: 1	Y:1		

4.5. Intelligent Agent Design

The intelligent agent is designed based on the behavior-oriented design method [63] with a modular structure, as illustrated in Figure 10. Each agent can naturally act on its own. First, the functional requirements of the agent are formulated based on the scenario analysis. Then, these requirements are decomposed into simple behavior. We use the five behavior types of the SPADE [52]: cyclic, one-shot, periodic, timeout, and finite state machine. The cyclic behavior is used for handling messages from humans or other agents. In SPADE, a cyclic behavior works as a "while loop" to be always ready to process incoming messages. SPADE also supports storing data in an "agent knowledge", a type of in-memory key-value store. After the message is handled, it is converted to structured data and is passed to the state behavior module. Different decision-making models (e.g., utility function, trained models, etc.) can be used to process the data depending on the agent's current state and settings. The output of the decision-making model is the activation of one-shot, timeout, or periodic behavior.

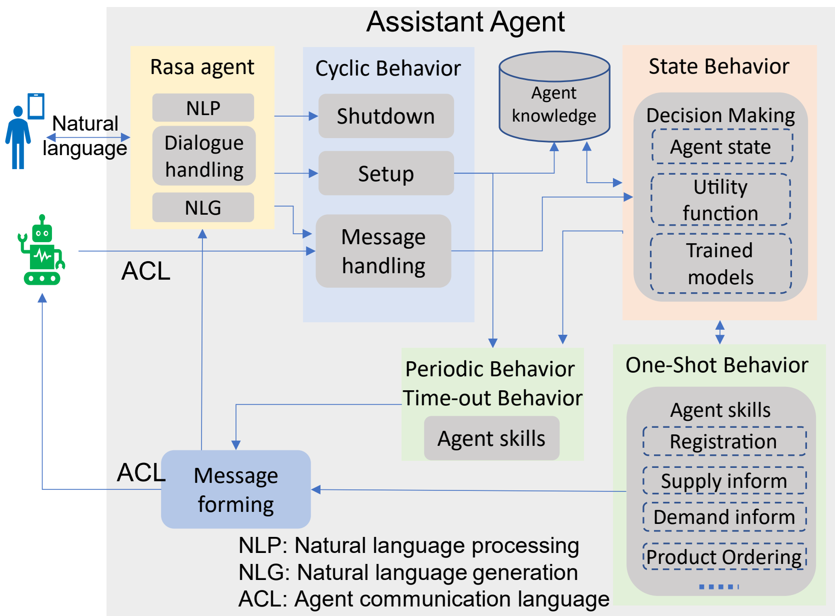


Figure 10. The illustration of the assistant agent component.

The structure of one-shot, periodic, and timeout behaviors are similar. The difference is that one-shot behavior can be run once at the activated time, that timeout behavior can be run once at a scheduled time, and that periodic behavior can be run many times in a scheduled period. "Agent skills" are developed as a data processing function and can be reused for all three types of these behaviors. The output data of the "agent skills" is formed into FIPA agent communication language (ACL) [68] and sent out to other agents or transformed into natural language before being sent to humans. We built a "Rasa agent" using the Rasa framework [70] to convert from natural language to ACL and vice versa. The Rasa agent will be specified in more detail in Section 7.

5. Multi-Agent Simulation

This section describes a multi-agent simulation using the "Simulation for MAS" approach presented in Section 2.5.2. Firstly, we develop ABM using the Mesa framework [64], and ABM works as a design-supported tool to develop CHASS. The objective of the simulation is to evaluate the functions and applicability of CHASS in a cooperative model in AFNs to understand the mechanism of resource allocation and negotiation, thereby helping to create new requirements for the evolution of CHASS. ABM is a simulation tool that supports the development of CHASS. When implementing CHASS in practice, ABM will be eliminated. Instead, users will use the mobile application to directly interact with their assistant agents using natural language similar to interacting with chatbots.

5.1. Simulation Model

Multi-agent simulation is an integral process of the development and evolution of CHASS. The intelligent agent will be designed, tested, evaluated, and improved through simulation. We use the Mesa framework [64] to build an ABM, including simulated agents, social structure (resource, rules), and environment. The simulated agents, which represent humans in the real world, have different attributes and decision-making models to represent the heterogeneous population. Each simulated agent in ABM is assisted by an intelligent agent in CHASS, and the communications between simulated agents and intelligent agents are done through the XMPP protocol. The ABM generates supply and demand data over time, and all generated data are transferred to the CHASS platform in every time step of the simulation process, as shown in Figure 11. After receiving data from ABM, CHASS starts four stages of service flow, as illustrated in Figure 6. After completing the operations in Stage 4, CHASS sends back a finished notification to inform ABM to start a new simulation step. There are two types of simulated agents for AFNs in ABM: customer and farmer. The supply data is generated depending on the farm capacity and farmer agents' crop decisions, and the demand data is generated based on customer agents' attributes and decisions.

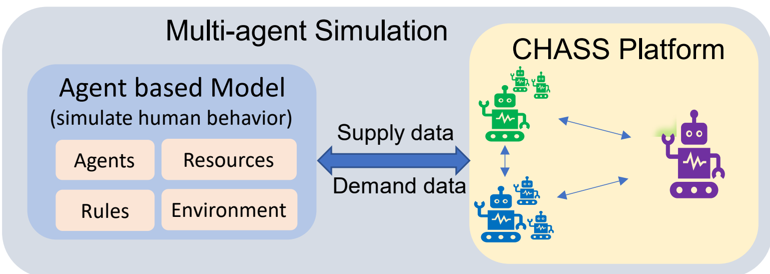


Figure 11. The illustration of the relationship between the ABM and the CHASS platform.

5.2. Develop Farmer Agent for Supply Data Generation

In ABM, each farmer agent owns a limited land size and chooses a group of products to grow. In practice, each farmer will have knowledge and experience for certain groups of products, and in the short term, they do not switch to other products. Therefore, in the simulation model, we assume that farmers will choose a group of products for the first crop and will not change to other products throughout the simulation. Each product type has land preparation time, harvest period, and product yield attributes. These parameters will remain constant during the simulation (Tables 2 and 3).

Table 2. Product parameters of the ABM model.

Product Group	Product Name	Yield per Acre (Tonne)	Harvest Period (Day)	Unit Weight (kg)	Preparation Time (Day)
(1) Lettuce group	Romaine	3.5–3.8	50–80	0.3	7–10
	Chinese lettuce	3.5–3.8	50–80	0.3	7–10
	Iceberg	3.5–3.8	50–80	0.3	7–10
	Escarole	3.5–3.8	50–80	0.3	7–10
(2) Cabbage group	Cabbage	7.0–7.5	60–90	0.5	7–10
	Cauliflower	7.0–7.5	60–90	0.5	7–10
	Chinese cabbage	7.0–7.5	60–90	0.5	7–10
	Broccoli	7.0–7.5	60–90	0.5	7–10

Table 3. Farmer agent parameters of the ABM model.

Parameter	Value	Explain
Land size	0.5 acre	Farmer-owned land size
Cultivated product group	(1) or (2)	Either the lettuce group or the cabbage group is selected for cultivation. In the simulation, half of the farmers grow product group 1, and the others grow product group 2.
Land cultivation ratio	100%	The percentage of cultivated land on owned land
The first crop start time	1~80 days	Time to start the first crop, randomly chosen
Farm split factor	4	Split cultivated land to multi-farm
Maximum harvest period	14 days	The period of harvest time of a crop.
Quantity harvest	0~max	Quantity is each harvest time—the maximum value is the unharvested quantity that remains on the farm.

In the simulation model, each farm is a simulated object with a life cycle (crop sowing, growth, harvesting, and ending) similar to an actual crop life cycle, as presented in Figure 12. At the crop preparation stage, farmers decide the cultivating product, farm size, and sowing time. Then, when the farm reaches harvest time, farmers make a harvest schedule (e.g., the number of harvests and the quantity of each harvest). Because of the different growth rates of vegetables, farmers can harvest a crop multiple times on the same farm to ensure a stable supply in AFNs. In this simulation, the maximum harvest period is set to 14 days. The supply capacity of the cooperative AFNs in each simulation time step depends on farm capacity and the farmer's harvest decision. After harvesting, the farmer agent must wait a period for land preparation before starting the next crop (creating a new farm object).

5.3. Developing a Customer Agent for Demand Data Generation

The entire demand is created by aggregating individual customer agent demand. Referring to the actual vegetable needs of each member in the cooperative model, the weekly demand of each customer agent is set up randomly from 4 to 10 units for each product type, equivalent to 1.2~3 kg lettuce (300 g/unit) and 2~5 kg cabbage (500 g/unit) consumption per week. Then, each customer can buy these products once or twice weekly. These parameters are set based on the buying behavior of co-op members in Taiwan. An important attribute of a customer agent is "demand flexibility" with True/False value. If the "demand flexibility" is set to *False*, it means the customer agent only wants to buy the preferred product and does not agree to switch to another product even if that product is in shortage. On the contrary, if "demand flexibility" is set to *True*, it means the customer agent may accept negotiation to switch to another product in the same product group (lettuce or cabbage) according to the customer's product preferences in a shortage situation. The customer parameters used to generate demand data are described in Table 4.

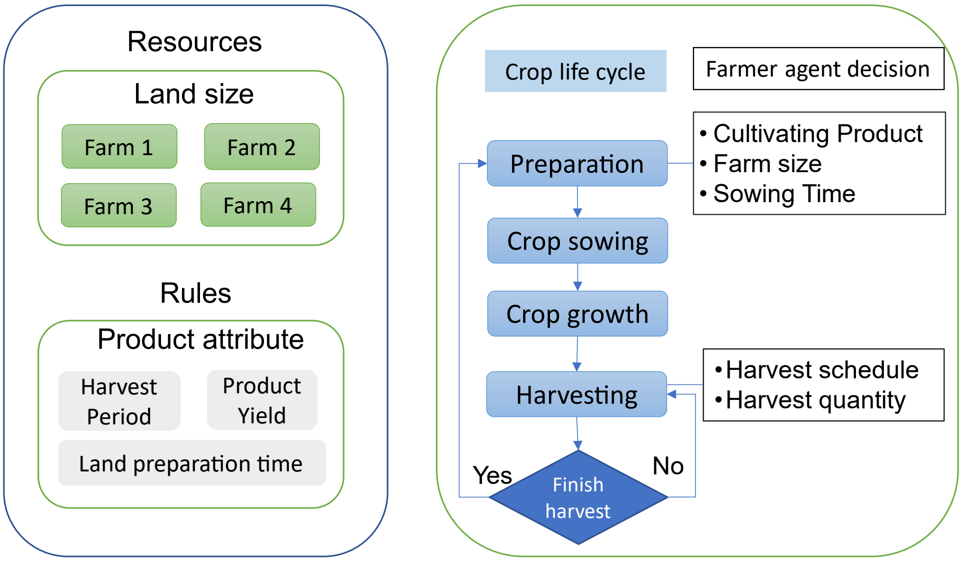


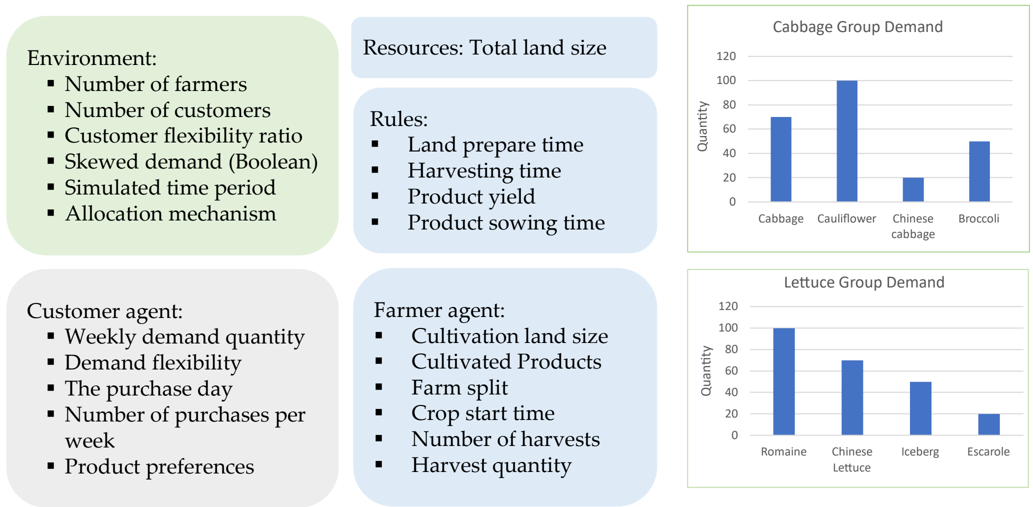
Figure 12. Resources, rules, and farmer agent decisions during the crop life cycle in the simulation.

Table 4. Customer agent parameters of ABM model.

Parameter	Value	Explain
Weekly demand quantity in unit weight	4~10	Depends on unit weight, e.g., lettuce (300 g/unit) or cabbage (500 g/unit) Random choice in the range 4~10, equivalent to 1.2–3 kg lettuce and 2–5 kg cabbage consumption per week
Purchases per week	1 or 2	The number of times customers purchase in a week
The purchase day	1~7	A choice of 1 or 2 days out of 7 days of the week
Demand flexibility (DF)	True/False	Customers agree or disagree with negotiation to switch to another product.
Product preferences (In case DF = False)	e.g., [cabbage]	Purchasing one product and not accepting a switch to another product
Product preferences (In case DF = True)	e.g., [Romaine, Iceberg, Chinese Lettuce, Escarole]	Purchasing one product in the product preferences lists in an orderly way and accepting negotiation to switch to another product

5.4. Simulation Model Parameters

Figure 13a provides an overview of all simulation parameters in ABM. These parameters can be divided into two groups: agent-level parameters (customers, farmers) and model-level parameters (environment, resources, and rules). The agent-level parameters can be set for each agent, and model-level parameters are set for the entire model. The supply data of each simulation step is generated based on land resources, crop rules, farmer's agent parameters, and the number of farmers. The demand data is generated based on the customer agent parameters, the demand flexibility ratio, and the skewed demand setting. Customer agents are classified into flexible customers. who accept negotiation. and non-flexible customers. The ratio of these two groups is set by the customer flexibility ratio, measured by the number of flexible customers over the total of customers. For example, if we assume that the simulation model has 400 customers and the demand flexibility ratio is 0.2, then 80 flexible customers will accept negotiation and the rest will not. The flexible customers will provide their demand preference (e.g., [romaine, iceberg, Chinese lettuce, escarole]), while non-flexible customers only provide one product demand (e.g., romaine). The skewed demand parameter (True or False) creates an imbalance in the product demand of the non-flexible customer group. The SciPy library [71] is used to generate a skew-normal distribution for the demand. Figure 13b illustrates the total demand of all non-flexible customers when the skewed demand is *True*. If the skewed demand is set to *False*, products will be assigned randomly to every non-flexible customer. As a result, there is only a slight imbalance in demand for different products.



(**a**) (**b**)

Figure 13. ABM simulation: (**a**) model parameters and (**b**) illustration of skewed demand.

6. Experimentation and Results

6.1. Experimental Setup

Multi-agent simulation is an integral part of the development of CHASS, in which the "design" and "evaluation through simulation" are integrated into a cyclic process. This subsection presents simulation results to evaluate the effect of the two-phase TTC-Negotiation mechanism mentioned in Section 4.4 on customer order fulfilled quantity. The "Fulfilled Quantity" is measured by the total quantity of all fulfilled orders after completing four stages of service flow in Figure 5 of simulation time steps.

In this simulation, we set up parameters for both CHASS and ABM to represent the real-world environment and the performance evaluation for CHASS (Figure 14a). First, we manipulate the resource allocation mechanism in CHASS: the two-phase TTC-Negotiation mechanism versus the first-come-first-served (*FCFS*) allocation method. Then, we generate demand and supply patterns in ABM by setting agent (customer, farmer) parameters, resources, rules, and environment (Tables 2–4 and Figure 14b). We set the food networks to have 10 farmers and 400 customers as the environmental parameters. The "skewed demand" parameter is set to *True* to simulate the imbalanced demand scenarios. The simulated period starts from time step 0 to 419 (420 days), in which the pre-harvest period is around 50 days, and harvesting is around 370 days, equivalent to one year of supply data. The "customer flexibility ratios" is set at 20 different levels (from 5% to 100%), with a 5% difference between the two adjacent levels. Simulation is performed two times for each "customer flexibility ratios" setting and each resource allocation mechanism. In summary, we performed 80 simulations, and each simulation included 369~370 data records (from day 50~51 to 419). The final simulation data consists of 29,560 records.

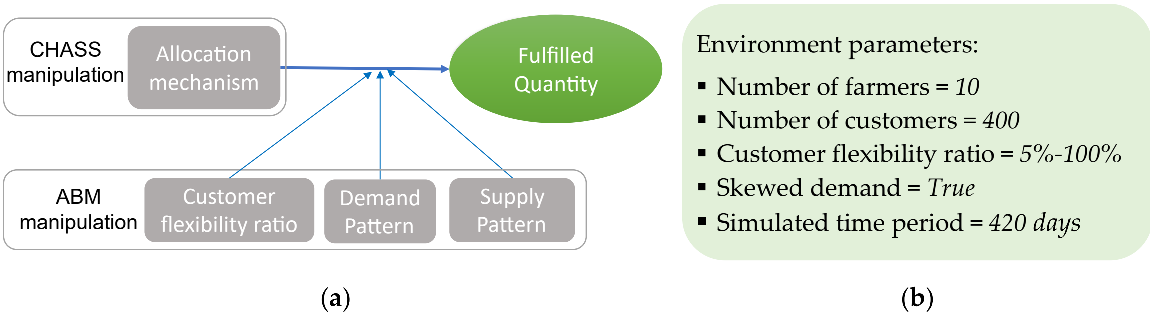


Figure 14. The illustration of parameters set for the simulation. (**a**) Simulation settings. (**b**) ABM environment parameters.

Figure 15a,b illustrate the supply pattern of the first 140 days. Depending on the farmer agent's decision on how many times to harvest, the supply can be as stable as Figure 15a or can show slight fluctuations as in Figure 15b. Figure 15c,d illustrate the demand pattern of the first simulation of 140 days. Demand has slight fluctuations as customers buy products one or two times on any day of the week. In addition, the demand for each product in the product group will be imbalanced since the "skewed demand" parameter is *True*.

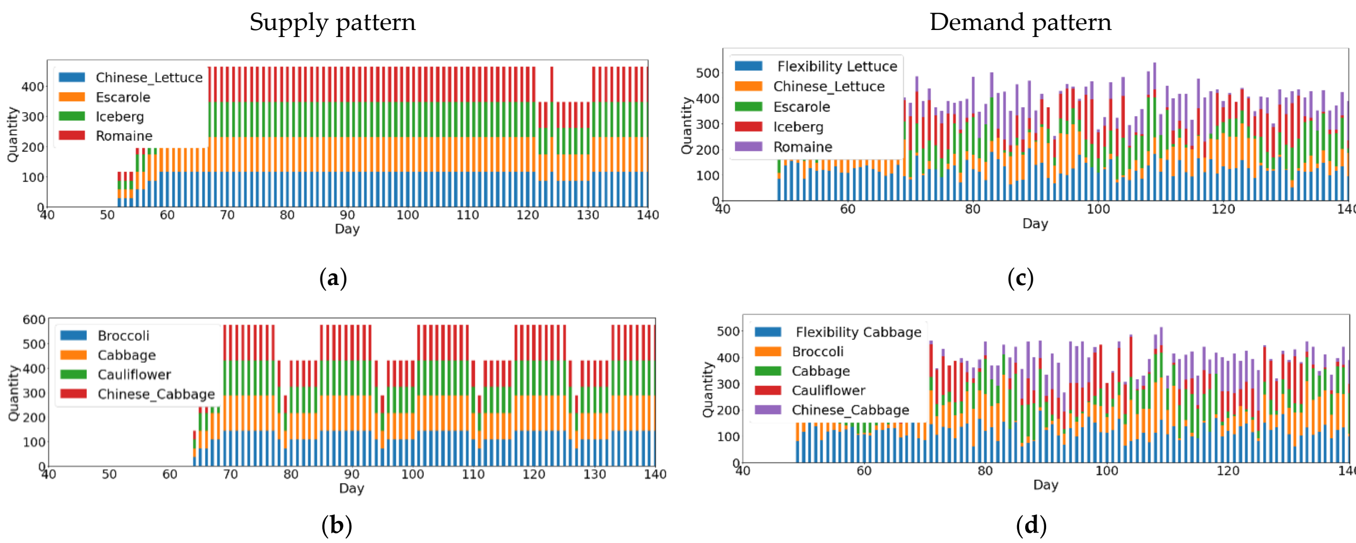


Figure 15. The illustration of demand and supply patterns generated by ABM. (**a**) supply pattern of the lettuce group. (**b**) supply pattern of the cabbage group. (**c**) demand pattern of the lettuce group. (**d**) demand patterns of the cabbage group.

6.2. First-Come-First-Served (FCFS) versus TTC-Negotiation Mechanism

Table 5 presents a simple example of the First-Come-First-Served (*FCFS*) method compared with the TTC-Negotiation mechanism. *FCFS* is a method of allocation based on time priority. As illustrated in Table 5, the demand of customers A, B, and C are allocated chronologically. A is allocated product X. The most preferred product of customer B is X, but X is out of stock, so B changed it to Y. Customer C is not allocated any products because he only wants to buy X. When using the TTC-Negotiation mechanism, the coordinator agent sends a negotiation message to ask A to switch to Z. If A agrees, the negotiation is successful. As a result, customer A will be allocated product Z, and C will be allocated product X.

Table 5. *FCFS* versus TTC-Negotiation mechanism.

Method	Time	Customer	Demand	Supply	Allocated
	0			X, Y, Z	
First Come, First Served	1	A	X (or Y, Z)	~~X~~, Y, Z	1
	2	B	X (or Y)	~~X~~, ~~Y~~, Z	1
	3	C	X	~~X~~, ~~Y~~, Z	0
			Total		2
TTC-Negotiation	4	A	X (or Y, Z)	X, ~~Y~~, ~~Z~~	1
	5	C	X	~~X~~, ~~Y~~, ~~Z~~	1
			Total		3

6.3. Performance Evaluation

We perform simulations with different supply and demand patterns and customer flexibility ratios to compare the allocated quantity between the *FCFS* method and the TTC-Negotiation mechanism. Two conclusions are drawn from the simulation:

1. Under the same condition of customer flexibility ratio, the allocation performance of the TTC-Negotiation mechanism is considerably moderated by the demand and supply pattern. This conclusion is clearly shown in Figure 16b. On some days, TTC-Negotiation performance is significantly higher than *FCFS* (e.g., day 95), while on other days, the difference is negligible (e.g., day 100).
2. The customer flexibility ratio (CFR) moderates the TTC-Negotiation mechanism performance. Figure 16 presents some simulation results at different scales of customer flexibility ratios. We started the simulation by setting CFR to 5% and found out that TTC-Negotiation obtained a slight improvement (Figure 16a) and continued to increase CFR. We saw significant improvement in the quantity made in transactions (Figure 16b); however, when the CFR is greater than 50%, we found that TTC-Negotiation no longer outperformed the *FCFS* method (Figure 16c,d).

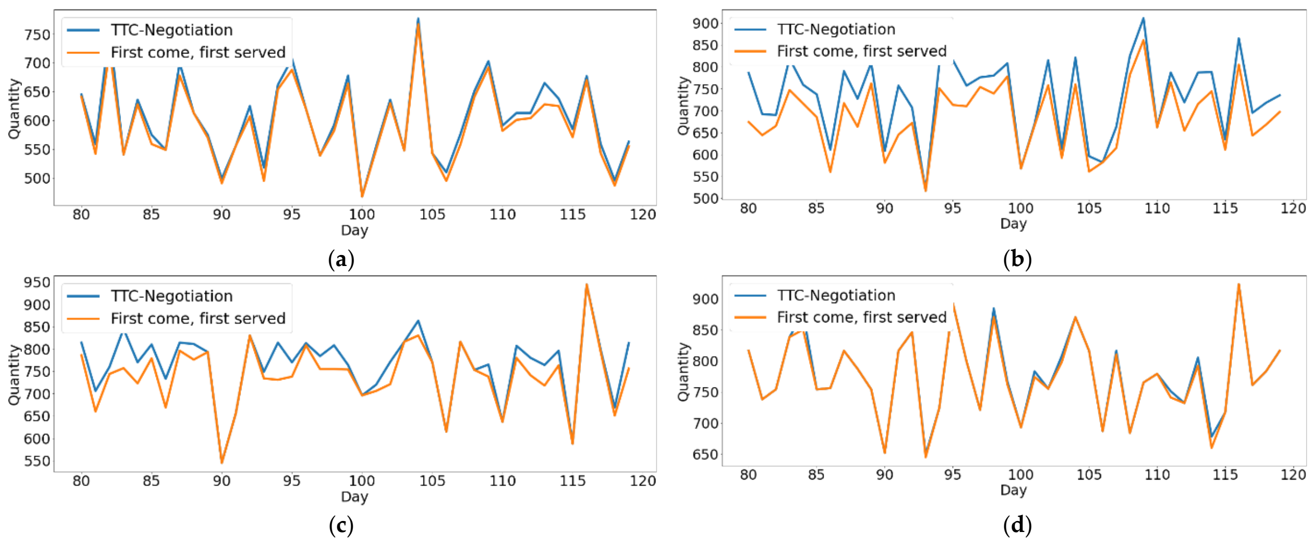
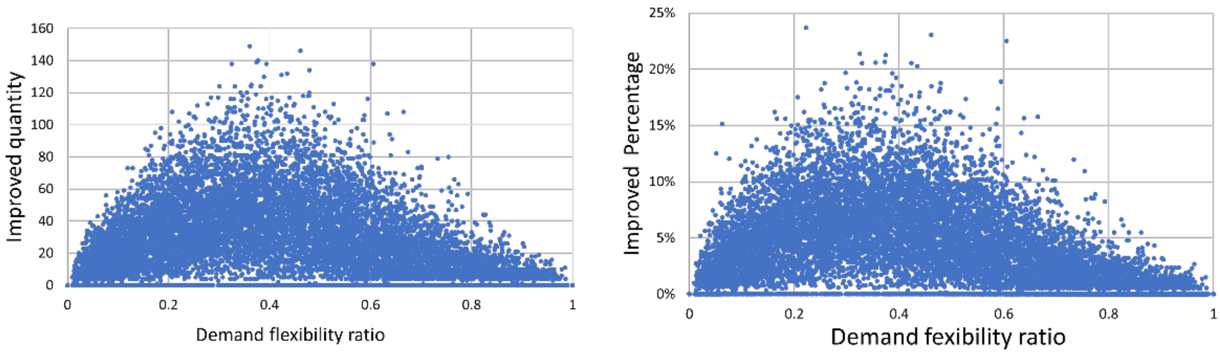


Figure 16. Evaluation of order fulfillment performance with different customer flexibility ratios. (**a**) Customer flexibility ratio CFR = 5%. (**b**) Customer flexibility ratio CFR = 30%. (**c**) Customer flexibility ratio CFR= 60%. (**d**) Customer flexibility ratio CFR = 90%.

To get a more general view of the effect of the customer flexibility ratio on allocation performance, we use a scatter plot to plot all cases with the Y-axis abbreviated as improvement in quantity (the difference of order fulfillment quantity) in Figure 17a and improvement in the percentage (improved quantity divided by the *FCFS* allocation quantity) in Figure 17b. Here, we define a new variable: "*demand flexibility ratio*" (*DFR*), Which is measured by the quantity of flexible demand over the total demand. The *DFR* is affected by CFR. For example, if we set CFR to 10% in simulation, *DFR* will fluctuate around 0.1, since the demand quantity of each customer is different. From the scatterplot, we can see that, if the *DFR* is in the range [0, 0.3], the *DFR* positively moderates the allocation performance. In contrast, if the *DFR* is in the range [0.4, 1], the *DFR* negatively moderates the allocation performance.



(**a**) (**b**)

Figure 17. Order fulfillment performance of the TTC-Negotiation mechanism versus *FCFS*. (**a**) Performance in quantity. (**b**) Performance in percentage.

6.4. The Analysis of Experimental Results

To better understand the influence of the TTC-Negotiation mechanism on the order fulfillment performance under the moderation of supply, demand patterns, and demand flexibility, we conducted regression analysis for all the simulation data following the conceptual framework illustrated in Figure 18. The independent variable (TTC-Negotiation) is a dummy variable that equals 0 if using the *FCFS* allocation method and 1 if using the TTC-Negotiation method. The dependent variable is the "order fulfillment performance" measured by the improvement of fulfilled quantity in percentage when applying the TTC-Negotiation mechanism over the *FCFS* method. In addition to the moderating variable *DFR* mentioned above, we define two new moderating variables: *minimum allocation quantity* (MinA) and *maximum allocation quantity* (MaxA), to represent the supply and demand patterns. For any supply and demand patterns, there is always a high and low threshold of allocation quantity for any allocation methods. The MinA happens when the *DFR* equals 0 (100% of customers are inflexible), and the MaxA happens when the *DFR* equals 1. Table 6 illustrates the meaning of these two variables through a simple example. With the same supply and demand pattern, the MinA and MaxA are 6 and 9, respectively.

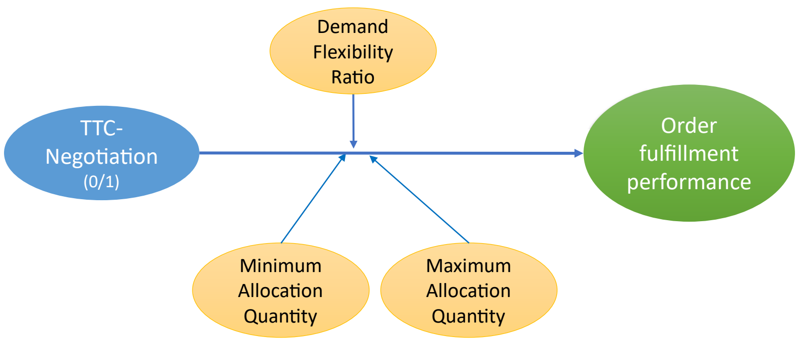


Figure 18. The conceptual framework of factors affects the order fulfillment performance.

Table 6. Illustration of the minimum allocation quantity and the maximum allocation quantity construct.

Product	**Minimum Allocation Quantity**			**Maximum Allocation Quantity**		
	Demand	**Supply**	**Allocated**	**Demand**	**Supply**	**Allocated**
Romaine	4	2	2	5 Romaine or iceberg	2	2
Iceberg	1	3	1		3	3
Cabbage	1	2	1	8 Cabbage or cauliflower	2	2
Cauliflower	5	2	2		2	2
	Total allocated quantity		6	Total allocated quantity		9

Since the moderation effect of *DFR* is opposite in different value ranges (Figure 17), we use the *piecewise-polynomial model* (also known as the *multiphase model*) to fit the data. The estimated model comprises two linear equations, each active over a different X range. The estimation equation has the form $Y = A + BX + C(X - D)SIGN(X - D)$, where D is the cut-off value that separates two linear equations. Using NCSS software from LLC, Kaysville, UT, USA, version 21.0.3 [72], we can estimate D = 0.31152 when the outcome performance is measured in percentage (Figure 17b). Based on this result, we use a cut-off value of *DFR*, 0.3, to separate the data into two groups and to run different regression models for each group. All standardized beta coefficients of the two regression models, as presented in Table 7, are significant at the 0.001 level. The interaction term between TTC-Negotiation and the *DFR* (Neg_x_DFR) is *0.732*, presenting the high positive moderation effect of the *DFR* in its range [0, 0.3], and -0.678 presented the negative moderation effect of the *DFR* in its range [0.3, 1].

Table 7. Regression analysis results.

Data Group	**Demand Flexibility Ratio $\leq$ 0.3**			**Demand Flexibility Ratio > 0.3**		
	Standardized Beta Coefficients	**t-Value**	***p*-Value**	**Standardized Beta Coefficients**	**t-Value**	***p*-Value**
Neg	−0.189	−6.955	−0.189	0.839	53.292	0.000
Neg_x_MinA	−1.393	−23.614	−1.393	−5.404	−112.129	0.000
Neg_x_MaxA	1.707	30.276	1.707	5.693	122.146	0.000
Neg_x_DFR	0.732	62.966	0.732	−0.678	−63.758	0.000
	R^2 = 0.673			R^2 = 0.766		

6.5. Formalization of Learning

All simulation results can be illustrated briefly, as shown in Figure 19. With any supply and demand pattern, we can determine two thresholds: minimum allocation quantity *MinA* and maximum allocation quantity *MaxA* ($MinA \leq MaxA$). If the overdemand or oversupply situation happens, its equivalents $MinA = MaxA$ and any allocation methods give the same order fulfillment performance. On the contrary, in case $MinA < MaxA$, if increasing *DFR* from 0, the performance of both the *FCFS* method and the TTC-Negotiation (TTCN) mechanism will increase, but TTCN performance increases at a higher speed, and it gives a better performance than *FCFS*. When the *DFR* crosses the cut-off value of 0.3, the TTCN performance is very close to *MaxA,* so it can no longer increase, while *FCFS* performance continues to increase following the increase of the *DFR*. When *DFR* equals 1, both *FCFS* and TTCN reach the *MaxA* value.

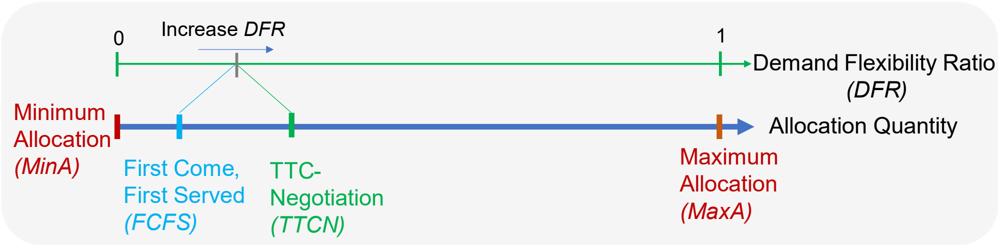


Figure 19. Formalizing the influence of demand flexibility on the order fulfillment performance.

7. Human–Agent Communication for CHASS Deployment

In Sections 5 and 6, we design the multi-agent simulation in which an ABM interacts with CHASS to evaluate the designed service system and formalize the learning. In this simulation, the simulated agents in ABM communicate to intelligent agents in CHASS through agent communication language (ACL). However, when implementing CHASS in practice, humans will communicate with assistant agents in CHASS through natural language. Human users will communicate with their assistants in CHASS in natural language via a chat application similar to interacting with commercial chatbots. The assistant agent can be installed on a personal computer or a cloud server. Users only need to install the chat application to be able to interact with their assistant agents. Furthermore, communication between human users and assistant agents is done through XMPP, an instant message protocol; therefore, users can also interact with other users on the same app by instant messages. The interaction between human–human, human–agent, and agent–agent is illustrated in Figure 20.

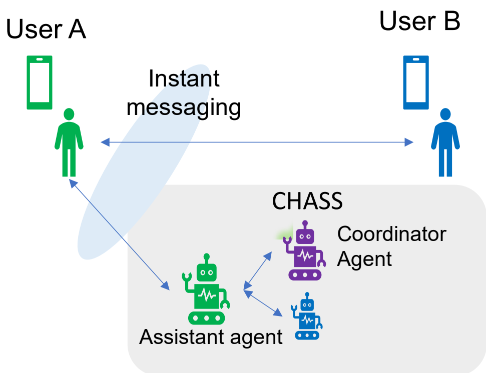


Figure 20. Human–agent communication.

To handle human–agent communication, the app needs to translate natural language to agent communication language and vice versa. To communicate with the assistant agent, users can use a mobile app that works as an XMPP client to send and receive instant messages. When the assistant agent receives a message in natural language, it will start the natural language processing (NLP) pipeline, as illustrated in Figure 21. The first step of natural language processing is tokenization. In this step, we use the spaCy library [73] for English and the Jieba library [74] for Chinese. Next, tokenized terms are passed through the feature process to convert them to vectors for user-intent classifier purposes. The user-intent classifier model was trained with a collection of human conversation data in pre-built scenarios. The classifier model will detect intent with a confidence score when receiving incoming messages. If the confidence score is less than the threshold set by users, that message will be classified as “out-of-scope”, and the model will reply with an out-of-scope response.

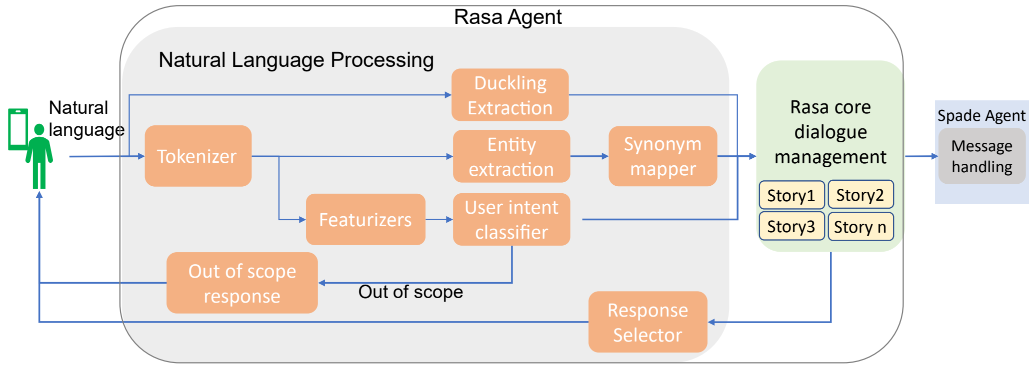


Figure 21. The natural language processing pipeline.

In parallel with the user-intent detection process, entities such as product names and locations will be extracted. We use the Duckling library [75] to detect and extract time, number, and weight units. Duckling supports multi-languages, so it can work for both English and Chinese. As a simple example, if a user inputs a sentence, " I want to buy 2 kg cabbage", the NLP module will detect the user's intent is "to buy", the product name "cabbage" is extracted, and Duckling will extract the number "2" and weight unit "kg". At the end of the process, the NLP converts natural language to structured data: [*"intent"* : *"buy"*, *"product"* : *"cabbage"*, *"quantity"* : 2, *"unit"* : "kg"]. Next, depending on user intent, the "Rasa core dialogue management" may respond in natural language to collect more information from human users. Finally, all structured data will be handled by SPADE cycle behavior, as mentioned in Figure 10.

8. Discussion and Conclusions

8.1. Summary

Concerned about the unsustainability of the conventional food systems, AFNs have emerged and are widely believed to be more sustainable. However, AFNs also face criticism for their limitations on economic sustainability. To increase economic efficiency, some AFNs formed an intermediary to aggregate products from producers and redistribute them to customers through a centralized digital platform to reduce transaction costs. However, such models will somewhat lose an essential feature of AFNs, which is the establishment of "strong relationships" between customers and producers, and a centralized platform is not technically sustainable.

This study develops a Cooperative Human–Agent Service System (CHASS) to overcome the limitations on the economic sustainability of AFNs and achieve technical sustainability based on decentralized technology. The outstanding features of CHASS are the human actor who has the agency for control and can be assisted by an assistant agent; a collaboration between humans, assistant agents, and coordination agents through the allocation and negotiation mechanism; and direct service exchange between actors.

8.2. Theoretical Contributions and Practical Implications

This research contributes to the existing literature in three aspects. *First*, this study proposes a two-phase TTC-Negotiation mechanism to facilitate food resource allocation and direct service exchange. In this study, we combine the "demand flexibility and resource allocation optimization" research stream that has proven to be economically beneficial and contributes to sustainable production and consumption in the energy field, and the research stream on "direct service exchange and value co-creation digital platform for sustainability" that contributes to the strengthening of human relationships and urban–rural linkages. The proposed two-phase TTC-Negotiation mechanism aims to balance the resource allocation optimization goal and human relationship establishment goal. *Second*, this study proposes a new approach to value co-creation through collaboration between humans and agents in a new generation of multi-agent systems that contribute to the human-in-the-loop research stream. *Third*, through the design, building, and evaluation process, this study contributes to service system design knowledge and evaluation methods by integrating ABM and MAS in multi-agent simulation.

This study was conducted in the context of Taiwan, which is affected by urbanization and rural decline, and the food system is oriented toward industrialized and globalized production. Therefore, the design of CHASS will make a practical contribution to generating new resources for expanding the AFNs model to reach a balance between AFNs and the conventional food system. Applying CHASS for cooperative AFNs can bring many benefits, such as reducing the cost of order processing, coordination, negotiation, and member communication, because many cooperatives in Taiwan still receive orders via a google form, ecommerce website, or Line chat and then process orders manually. Furthermore, by better food allocation and direct service exchange, CHASS can help strengthen human relationships and contribute to urban–rural linkage. In addition, CHASS is designed to be an open and distributed system to achieve economic and technical sustainability to ensure long-term benefits to the community.

8.3. Limitations and Future Research Directions

In this study's simulation, we assume that CHASS is applied in a small cooperative in AFNs, and prices are set at a reasonable level based on transparency, similar to the price-setting mechanism in the CSA model. Therefore, we ignore the influence of the price factor on customer choice and only focus on evaluating the allocation mechanism and customer demand flexibility. However, if CHASS is deployed in AFNs with many customers and producers, the price factor certainly affects supply and demand. Therefore, developing a price consensus mechanism between members and including price factors in the customer decision model is necessary for future work. Furthermore, in Phase 1 of the TTC-Negotiation mechanism of the simulation, product supply is allocated to demand based on demand time priority. In future research, the TTC-Negation mechanism can be evaluated based on combining other priority methods such as Recency, Frequently, Monetary (RFM) models [69].

The second limitation is related to the capacity of the assistant agent to decide on behalf of the human. In the first CHASS prototype, the assistant agent can only make decisions based on a preset utility function. The user needs to set the parameters for the utility function or decision model. This issue will significantly affect the feasibility of practical deployment. In the next version, an assistant agent can be developed to be able to self-adjust the decision-making model (e.g., self-adjust utility function parameters) or self-build the decision-making model through the reinforcement learning process.

Another limitation is the flexibility level of assistant agents in communication with humans. In this study, we only prepared limited scenarios to train the assistant agent to detect user intents (e.g., buy or sell). Therefore, if the user asks other questions, they are classified as "out of scope, " which will probably make the user uncomfortable. We plan to develop a shared knowledge base so that all agents can access and retrieve data to handle questions related to sustainability from users.

Although CHASS offers many advantages and promises to contribute much to sustainability, more research needs to be done regarding CHASS implementation for a specific AFN model. Due to limited resources, in this study, the assistant agent still has several limitations on communication and the decision-making model. The ability to communicate with humans in natural language and self-adjusted decision-making models through communication are two essential features of the assistant agent expected to be implemented in CHASS in the near future to realize the human–agent cooperative service system.

Author Contributions: Conceptualization, F.-R.L.; Data curation, V.-C.T.; Formal analysis, V.-C.T.; Methodology, F.-R.L.; Project administration, F.-R.L.; Resources, F.-R.L.; Software, V.-C.T.; Supervision, F.-R.L.; Validation, V.-C.T.; Visualization, V.-C.T.; Writing—original draft, V.-C.T.; Writing—review & editing, F.-R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, Republic of China, grant number 109-2410-H-007-019-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Edwards, F. Alternative food networks. In *Encyclopedia of Food and Agricultural Ethics*; Springer: Dordrecht, The Netherlands, 2016; pp. 1–7. [CrossRef]
2. Forssell, S.; Lankoski, L. The sustainability promise of alternative food networks: An examination through "alternative" characteristics. *Agric. Hum. Values* **2015**, *32*, 63–75. [CrossRef]
3. Milestad, R.; Westberg, L.; Geber, U.; Björklund, J. Enhancing adaptive capacity in food systems: Learning at farmers markets in Sweden. *Ecol. Soc.* **2010**, *15*, 29. [CrossRef]
4. Goodman, D.; DuPuis, E.M.; Goodman, M.K. *Alternative Food Networks: Knowledge, Practice, and Politics*; Routledge: Abingdon-on-Thames, UK, 2012.
5. Arkko, J. The influence of internet architecture on centralised versus distributed internet services. *J. Cyber Policy* **2020**, *5*, 30–45. [CrossRef]
6. Khoshafian, S. Can the Real Web 3.0 Please Stand Up? Available online: https://www.rtinsights.com/can-the-real-web-3-0-please-stand-up/ (accessed on 2 December 2021).
7. Vargo, S.L.; Lusch, R.F. *The SAGE Handbook of Service-Dominant Logic*; SAGE Publications Limited: Thousand Oaks, CA, USA, 2018.
8. Lusch, R.F.; Vargo, S.L.; Wessels, G. Toward a conceptual foundation for service science: Contributions from service-dominant logic. *IBM Syst. J.* **2008**, *47*, 5–14. [CrossRef]
9. Morgan, K.; Marsden, T.; Murdoch, J. *Worlds of Food: Place, Power, and Provenance in the Food Chain*; Oxford University Press on Demand: Oxford, UK, 2008.
10. Renting, H.; Marsden, T.K.; Banks, J. Understanding alternative food networks: Exploring the role of short food supply chains in rural development. *Environ. Plan. A* **2003**, *35*, 393–411. [CrossRef]
11. Lee, R. Shelter from the storm? Geographies of regard in the worlds of horticultural consumption and production. *Geoforum* **2000**, *31*, 137–157. [CrossRef]
12. Feagan, R.; Henderson, A. Devon Acres CSA: Local struggles in a global food system. *Agric. Hum. Values* **2009**, *26*, 203–217. [CrossRef]
13. Marsden, T.; Banks, J.; Bristow, G. Food supply chain approaches: Exploring their role in rural development. *Sociol. Rural.* **2000**, *40*, 424–438. [CrossRef]
14. Guthman, J.; Morris, A.W.; Allen, P. Squaring farm security and food security in two types of alternative food institutions. *Rural Sociol.* **2006**, *71*, 662–684. [CrossRef]
15. Gomiero, T.; Pimentel, D.; Paoletti, M.G. Environmental impact of different agricultural management practices: Conventional vs. organic agriculture. *Crit. Rev. Plant Sci.* **2011**, *30*, 95–124. [CrossRef]
16. Schönhart, M.; Penker, M.; Schmid, E. Sustainable local food production and consumption: Challenges for implementation and research. *Outlook Agric.* **2009**, *38*, 175–182. [CrossRef]
17. Born, B.; Purcell, M. Avoiding the local trap: Scale and food systems in planning research. *J. Plan. Educ. Res.* **2006**, *26*, 195–207. [CrossRef]
18. Esquinas-Alcázar, J. Protecting crop genetic diversity for food security: Political, ethical and technical challenges. *Nat. Rev. Genet.* **2005**, *6*, 946–953. [CrossRef]
19. Reisch, L.; Eberle, U.; Lorek, S. Sustainable food consumption: An overview of contemporary issues and policies. *Sustain. Sci. Pract. Policy* **2013**, *9*, 7–25. [CrossRef]
20. Sundkvist, Å.; Milestad, R.; Jansson, A. On the importance of tightening feedback loops for sustainable development of food systems. *Food Policy* **2005**, *30*, 224–239. [CrossRef]
21. Kloppenburg, J.; Hendrickson, J.; Stevenson, G.W. Coming in to the foodshed. *Agric. Hum. Values* **1996**, *13*, 33–42. [CrossRef]
22. James, S.W. Beyond local food: How supermarkets and consumer choice affect the economic viability of small-scale family farms in Sydney, Australia. *Area* **2016**, *48*, 103–110. [CrossRef]
23. Coley, D.; Howard, M.; Winter, M. Local food, food miles and carbon emissions: A comparison of farm shop and mass distribution approaches. *Food Policy* **2009**, *34*, 150–155. [CrossRef]
24. Virtanen, Y.; Kurppa, S.; Saarinen, M.; Katajajuuri, J.-M.; Usva, K.; Mäenpää, I.; Mäkelä, J.; Grönroos, J.; Nissinen, A. Carbon footprint of food–approaches from national input–output statistics and a LCA of a food portion. *J. Clean. Prod.* **2011**, *19*, 1849–1856. [CrossRef]
25. Weber, C.L.; Matthews, H.S. Food-miles and the relative climate impacts of food choices in the United States. *Environ. Sci. Technol.* **2008**, *42*, 3508–3513. [CrossRef]
26. Adam, K.L. *Community Supported Agriculture*; ATTRA-National Sustainable Agriculture Information Service: Butte, MT, USA, 2006.

27. Woods, T.; Ernst, M.; Tropp, D. *Community Supported Agriculture: New Models for Changing Markets*; United States Department of Agriculture, Agricultural Marketing Service: Washington, DC, USA, 2017.
28. Freedman, M.R.; King, J.K. Examining a new "pay-as-you-go" community-supported agriculture (CSA) model: A case study. *J. Hunger. Environ. Nutr.* **2016**, *11*, 122–145. [CrossRef]
29. Gebre, T.; Gebremedhin, B. The mutual benefits of promoting rural-urban interdependence through linked ecosystem services. *Glob. Ecol. Conserv.* **2019**, *20*, e00707. [CrossRef]
30. Preiss, P.; Charão-Marques, F.; Wiskerke, J. Fostering Sustainable Urban-Rural Linkages through Local Food Supply: A Transnational Analysis of Collaborative Food Alliances. *Sustainability* **2017**, *9*, 1155. [CrossRef]
31. Akaka, M.A.; Vargo, S.L. Technology as an operant resource in service (eco) systems. *Inf. Syst. E-Bus. Manag.* **2014**, *12*, 367–384. [CrossRef]
32. Vargo, S.L.; Lusch, R.F. Service-dominant logic: What it is, what it is not, what it might be. In *The Service-Dominant Logic of Marketing*; Routledge: Abingdon-on-Thames, UK, 2015; pp. 43–56.
33. Vargo, S.L.; Lusch, R.F. It's all B2B … and beyond: Toward a systems perspective of the market. *Ind. Mark. Manag.* **2011**, *40*, 181–187. [CrossRef]
34. Lusch, R.F.; Vargo, S.L.; Tanniru, M. Service, value networks and learning. *J. Acad. Mark. Sci.* **2010**, *38*, 19–31. [CrossRef]
35. Głąbska, D.; Skolmowska, D.; Guzek, D. Food Preferences and Food Choice Determinants in a Polish Adolescents' COVID-19 Experience (PLACE-19) Study. *Nutrients* **2021**, *13*, 2491. [CrossRef]
36. Wadolowska, L.; Babicz-Zielinska, E.; Czarnocinska, J. Food choice models and their relation with food preferences and eating frequency in the Polish population: POFPRES study. *Food Policy* **2008**, *33*, 122–134. [CrossRef]
37. Köster, E.P. Diversity in the determinants of food choice: A psychological perspective. *Food Qual. Prefer.* **2009**, *20*, 70–82. [CrossRef]
38. Forouli, A.; Bakirtzis, E.A.; Papazoglou, G.; Oureilidis, K.; Gkountis, V.; Candido, L.; Ferrer, E.D.; Biskas, P. Assessment of Demand Side Flexibility in European Electricity Markets: A Country Level Review. *Energies* **2021**, *14*, 2324. [CrossRef]
39. Shapley, L.; Scarf, H. On cores and indivisibility. *J. Math. Econ.* **1974**, *1*, 23–37. [CrossRef]
40. Ma, J. Strategy-proofness and the strict core in a market with indivisibilities. *Int. J. Game Theory* **1994**, *23*, 75–83. [CrossRef]
41. Abdulkadiroğlu, A.; Sönmez, T. School choice: A mechanism design approach. *Am. Econ. Rev.* **2003**, *93*, 729–747. [CrossRef]
42. Su, X.; Zenios, S.A. Recipient choice can address the efficiency-equity trade-off in kidney transplantation: A mechanism design model. *Manag. Sci.* **2006**, *52*, 1647–1660. [CrossRef]
43. Kesten, O. Coalitional strategy-proofness and resource monotonicity for house allocation problems. *Int. J. Game Theory* **2009**, *38*, 17–21. [CrossRef]
44. Schummer, J.; Vohra, R.V. Assignment of arrival slots. *Am. Econ. J. Microecon.* **2013**, *5*, 164–185. [CrossRef]
45. Dur, U.; Ünver, M.U. Two-sided matching via balanced exchange: Tuition and worker exchanges. *J. Political Econ.* **2019**, *127*, 1156–1177. [CrossRef]
46. Wooldridge, M. *An Introduction to Multiagent Systems*; John Wiley & Sons Ltd.: Chichester, West Sussex, UK, 2009.
47. Davis, R. *Report on the Workshop on Distributed AI*; Massachusetts Institute of Technology, Artificial Intelligence Laboratory: Cambridge, MA, USA, 1980.
48. Knoeri, C.; Binder, C.R.; Althaus, H.-J. An agent operationalization approach for context specific agent-based modeling. *J. Artif. Soc. Soc. Simul.* **2011**, *14*, 1729. [CrossRef]
49. Niazi, M.; Hussain, A. Agent-based computing from multi-agent systems to agent-based models: A visual survey. *Scientometrics* **2011**, *89*, 479–499. [CrossRef]
50. Michel, F.; Ferber, J.; Drogoul, A. Multi-agent systems and simulation: A survey from the agent community's perspective. In *Multi-Agent Systems*; CRC Press: Boca Raton, FL, USA, 2009; pp. 17–66.
51. Chun, H.W.; Wong, R.Y. N∗—An agent-based negotiation algorithm for dynamic scheduling and rescheduling. *Adv. Eng. Inform.* **2003**, *17*, 1–22. [CrossRef]
52. Palanca, J.; Terrasa, A.; Julian, V.; Carrascosa, C. SPADE 3: Supporting the New Generation of Multi-Agent Systems. *IEEE Access* **2020**, *8*, 182537–182549. [CrossRef]
53. Billhardt, H.; Julián, V.; Corchado, J.M.; Fernández, A. An architecture proposal for human-agent societies. In Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, Salamanca, Spain, 4–6 June 2014; pp. 344–357.
54. González-Briones, A.; De La Prieta, F.; Mohamad, M.; Omatu, S.; Corchado, J. Multi-Agent Systems Applications in Energy Optimization Problems: A State-of-the-Art Review. *Energies* **2018**, *11*, 1928. [CrossRef]
55. Reis, I.F.G.; Gonçalves, I.; Lopes, M.A.R.; Antunes, C.H. A multi-agent system approach to exploit demand-side flexibility in an energy community. *Util. Policy* **2020**, *67*, 101114. [CrossRef]
56. Damacharla, P.; Dhakal, P.; Bandreddi, J.P.; Javaid, A.Y.; Gallimore, J.J.; Elkin, C.; Devabhaktuni, V.K. Novel Human-in-the-Loop (HIL) Simulation Method to Study Synthetic Agents and Standardize Human–Machine Teams (HMT). *Appl. Sci.* **2020**, *10*, 8390. [CrossRef]
57. Bosse, S.; Engel, U. Real-Time Human-In-The-Loop Simulation with Mobile Agents, Chat Bots, and Crowd Sensing for Smart Cities. *Sensors* **2019**, *19*, 4356. [CrossRef]
58. Lusch, R.F.; Nambisan, S. Service innovation: A service-dominant logic perspective. *MIS Q.* **2015**, *39*, 155–175. [CrossRef]

59. Yao, G.; Miao, J. Service Value Co-Creation in Digital Platform Business: A Case of Xianyu Idle Trading Platform. *Sustainability* **2021**, *13*, 11296. [CrossRef]
60. Espelt, R. Agroecology prosumption: The role of CSA networks. *J. Rural Stud.* **2020**, *79*, 269–275. [CrossRef]
61. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design science in information systems research. *MIS Q.* **2004**, *28*, 75–105. [CrossRef]
62. Padgham, L.; Winikoff, M. *Prometheus: A Methodology for Developing Intelligent Agents*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 174–185.
63. Bryson, J.J. *The Behavior-Oriented Design of Modular Agent Intelligence*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 61–76.
64. Masad, D.; Kazil, J. MESA: An agent-based modeling framework. In Proceedings of the 14th PYTHON in Science Conference, Austin, TX, USA, 6–12 July 2015; pp. 53–60.
65. Maghsudi, S.; Stańczak, S. Hybrid centralized–distributed resource allocation for device-to-device communication underlaying cellular networks. *IEEE Trans. Veh. Technol.* **2015**, *65*, 2481–2495. [CrossRef]
66. Consoli, A.; Tweedale, J.; Jain, L. The link between agent coordination and cooperation. In Proceedings of the International Conference on Intelligent Information Processing, Adelaide, Australia, 20–23 September 2006; pp. 11–19.
67. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP; RFC 6122, RFC series (ISSN 2070-1721), CA, USA, March 2011. Available online: https://www.rfc-editor.org/rfc/rfc6122 (accessed on 30 March 2022).
68. Committee, I.F.S. *FIPA Communicative Act Library Specification*; Technical Report; Foundation for Intelligent Physical Agents: Geneva, Switzerland, 2000.
69. Birant, D. Data mining using RFM analysis. In *Knowledge-Oriented Applications in Data Mining*; IntechOpen: Rijeka, Croatia, 2011.
70. Bocklisch, T.; Faulkner, J.; Pawlowski, N.; Nichol, A. Rasa: Open source language understanding and dialogue management. *arXiv* **2017**, arXiv:1712.05181.
71. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef] [PubMed]
72. Hintze, J.L. *User's Guide III: Regression and Curve Fitting*; NCSS Statistical Software: Kaysville, UT, USA, 2007.
73. Explosion Inc. A. spaCy: Industrial-Strength Natural Language Processing in Python. Available online: https://github.com/explosion/spaCy (accessed on 30 March 2022).
74. Sun, J. Jieba: Chinese Text Segmentation. Available online: https://github.com/fxsjy/jieba (accessed on 30 March 2022).
75. Facebook Inc. Duckling: Haskell Library that Parses Text into Structured Data. Available online: https://github.com/facebook/duckling (accessed on 30 March 2022).