future internet

Article

An Intelligent System to Ensure Interoperability for the Dairy Farm Business Model

Adina Cretan [1,*], Cristina Nica [1], Carlos Coutinho [2], Ricardo Jardim-Goncalves [3] and Ben Bratu [4]

[1] Department of Economic Sciences, Faculty of Economics and Business Administration, "Nicolae Titulescu" University of Bucharest, 040051 Bucharest, Romania; nica.cristina87@gmail.com

[2] Caixa Mágica Software/Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, 1649-026 Lisbon, Portugal; carlos.coutinho@caixamagica.pt

[3] CTS, Departamento de Engenharia Electrotecnica, FCT University NOVA Lisbon, UNINOVA, 2829-516 Lisbon, Portugal; rg@uninova.pt

[4] Extreme Computing R&D High Level Software Team, Datadirect Networks, R&D, 92360 Meudon, France; bbratu@yahoo.com

* Correspondence: bratuadina@gmail.com

Abstract: Picking reliable partners, negotiating synchronously with all partners, and managing similar proposals are challenging tasks for any manager. This challenge is even harder when it concerns small and medium enterprises (SMEs) who need to deal with short budgets and evident size limitations, often leading them to avoid handling very large contracts. This size problem can only be mitigated by collaboration efforts between multiple SMEs, but then again this brings back the initially stated issues. To address these problems, this paper proposes a collaborative negotiation system that automates the outsourcing part by assisting the manager throughout a negotiation. The described system provides a comprehensive view of all negotiations, facilitates simultaneous bilateral negotiations, and provides support for ensuring interoperability among multiple partners negotiating on a task described by multiple attributes. In addition, it relies on an ontology to cope with the challenges of semantic interoperability, it automates the selection of reliable partners by using a lattice-based approach, and it manages similar proposals by allowing domain experts to define a satisfaction degree for each SME. To showcase this method, this research focused on small and medium-size dairy farms (DFs) and describes a negotiation scenario in which a few DFs are able to assess and generate proposals.

Keywords: multi-agent system; collaborative negotiation system; formal concept analysis; ontology; dairy farm; semantic interoperability

check for updates

Citation: Cretan, A.; Nica, C.; Coutinho, C.; Jardim-Goncalves, R.; Bratu, B. An Intelligent System to Ensure Interoperability for the Dairy Farm Business Model. *Future Internet* **2021**, *13*, 153. https://doi.org/10.3390/fi13060153

Academic Editors: Stefano Modafferi and Francesco Lelli

Received: 2 May 2021
Accepted: 9 June 2021
Published: 12 June 2021

1. Introduction

Small and medium-size enterprises (SMEs) are considered the backbone of the economy in many countries [1–3]. Indeed, SMEs play an important role in creating jobs, and promoting innovation [4,5]. Though SMEs substantially contribute to the economy, they tend to be underrepresented in the negotiation of large contracts [6]. This situation is mainly because of the lack of resources, which are required when engaging in large contracts [7,8]. Therefore, customers perceive SMEs as too risky and weaker counterparts for many large market players. To address this problem, collaborative partnerships with local SMEs can be formed in order to outsource work [9]. However, choosing which partners to collaborate with is a critical decision for many managers. In addition, trying to negotiate synchronously with all partners in a traditional manner to avoid losing better collaborations is a time- and resource-consuming task.

To assist SMEs in outsourcing the work and making proper decisions, a multi-agent system (MAS) can be used. In [10], MAS was defined as a loosely coupled network of problem-solving agents that seek together to find answers to issues that are beyond the particular capabilities or knowledge of each agent. In [11], Kumari et al. proposed in a self-adaptive multi-agent architecture to address the outsourcing task, where the selection of suitable partners is made by merely querying a database.

Regarding this approach, the major challenge for SMEs is to realize and preserve collaboration among partners at the business level and the information technology (IT) level that requires new ideas to handle the diverse nature of systems or applications, platforms, and infrastructures.

In this respect, Ghobakhloo et al. [12] highlighted the major position of the IT adoption process in SMEs to support their businesses. The authors propose a conceptual framework of effective IT adoption, as well as a strategy for SMEs to succeed in IT institutionalization at different adoption stages.

In dynamic business networks, a crucial challenge of the involved organizations is to achieve and maintain their information systems' interoperability in relation to continuous changes. Addressing this problem, Vrchota and Řehoř [13] emphasized the importance of collaborative work as a vital solution for improving the ability of SMEs to adapt to the evolving environment, innovate, and compete.

Emphasizing the same needs of SMEs to improve resource efficiency, increase competitiveness, and access new markets, Rizos et al. [14] identified various barriers that prevent SMEs from implementing new business models, such as a lack of technical skills and a lack of financial resources.

To tackle this issue in previous studies [15,16], the authors devised a collaborative negotiation system that automates the outsourcing part by assisting a manager throughout a negotiation. Briefly, Cretan et al. [15] proposed an MAS that: (1) provides a comprehensive view of all negotiations, (2) facilitates simultaneous bilateral negotiations, and (3) provides support for ensuring interoperability among multiple partners negotiating on a task described by multiple attributes.

This article focuses on small and medium-size dairy farms (DFs) that have to deal with challenging contracts by outsourcing the work to reliable partners. Indeed, the consumption of dairy products nowadays has increased (https://www.grandviewresearch.com/industry-analysis/dairy-product-market (accessed on 9 June 2021)) due to customers that have shifted their preferences from meat to dairy products. A *reliable partner* is a DF that complies with the negotiation manager standards, and has a high satisfaction degree measured based on the overall experience with the DF. To come to the aid of a DF manager, we rely on the collaborative negotiation system proposed in [15]. Here, we enhance this system as follows:

- To cope with the challenges of semantic interoperability, DF ontology is shared by all involved negotiation agents. This ontology facilitates agent communication by providing a shared vocabulary.
- To select reliable partners, an automated lattice-based component is proposed. This component relies on data about DFs gathered from catalogues (online or printed) and/or records of past deals. Formal concept analysis (FCA; [17]) is used for data analysis and to build a lattice of concepts (i.e., a search space of partners).
- To manage similar proposals that occur during the negotiation, a satisfaction degree is defined for each possible partner. This measure is defined by domain experts based on various characteristics (e.g., the location of DF and the type of milk).

The rest of the paper is organised as follows. Section 2 introduces a DF scenario. Related work is discussed in Section 3. Section 4 gives the technical instrumentation of the presented work. Section 5 formalises and illustrates the collaborative negotiation system. Section 6 presents a negotiation scenario. Section 7 concludes the paper and highlights some directions for future work.

2. Motivating Scenario

Small and medium-sized DFs have to collaborate in supplying significant quantities of dairy products. Consequently, a small/medium-sized DF can strengthen its position in the marketplace by winning and honouring challenging contracts.

Let us consider a medium-sized DF^M that got a contract to supply daily a considerable quantity of yoghurt to 100 schools from the north-west region of Romania. To fulfill the contract, DF^M has to outsource the work to local small-sized DFs (i.e., partners). To this end, DF^M may find and negotiate with reliable partners in a traditional manner. For example, DF^M has to look for partners and to gather data from catalogues and/or records of past deals; to select only the suitable ones based on, e.g., the employed feed or the type of milk; and to negotiate asynchronously with each partner. Table 1 shows some example DF data; we considered six DFs, where each DF was described with three key characteristics: the type of food used to feed animals (i.e., natural forage and growth stimulant), the type of milk (i.e., pasteurised and unpasteurised), and the DF's surroundings (i.e., forests and industrial areas). However, by following the traditional manner, DF^M has to deal with various time-consuming tasks, and, regardless, there is a lack of a global view of the negotiation process. As a result, DF^M is overwhelmed by the outsourcing part and can make hasty decisions.

Table 1. Example DF data.

Dairy Farm ID	Animal Food	Type of Milk	Surroundings
DF1	Natural forage	Pasteurised	Forests and Industrial areas
DF2	Natural forage and Growth stimulant	Pasteurised	Forests
DF3	Growth stimulant	Unpasteurised	Industrial areas
DF4	Natural forage	Pasteurised and Unpasteurised	Forests
DF5	Growth stimulant	Unpasteurised	Industrial areas
DF6	Natural forage	Pasteurised and Unpasteurised	Forests and Industrial areas

To avoid the pitfalls of the traditional manner, small and medium-sized DFs can automate the outsourcing part by using the negotiation system presented in [18,19]. For instance, this collaborative system assists DF^M throughout the negotiation by providing a comprehensive view of all negotiations and facilitating simultaneous bilateral negotiations. In addition, the proposed collaborative negotiation system provides support for ensuring interoperability among multiple partners negotiating on a task described by multiple attributes.

Therefore, in the negotiation scenario, one can employ the DF ontology illustrated in Figure 1. For instance, DF^M requests the price of ovine dairy products, and the partners return the price of goat and/or sheep dairy products according to the presented ontology. Additionally, we propose an automated lattice-based component for picking partners; and we define a satisfaction degree for each partner to be able to choose among similar proposals.

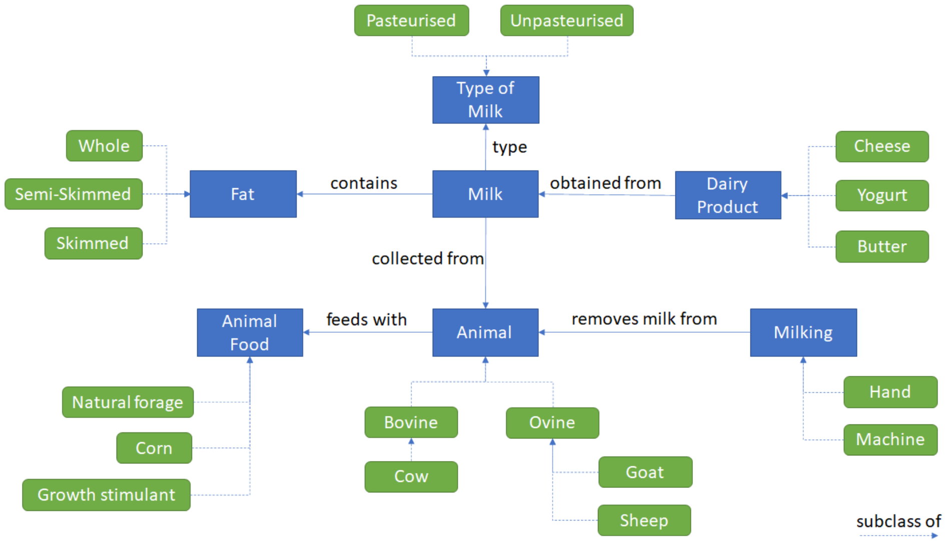


Figure 1. DF ontology.

3. Related Work

Communication is indispensable in distributed MAS, so many research papers have proposed ontology-based agent communication [20–22]. In this respect, Malucelli and da Costa Oliveira [23] introduced ontology-services in order to enhance the semantic interoperability of agents. Similarly, in this work, an ontology is employed to provide a vocabulary shared by all the agents involved in the negotiation process. In the same context, Arora and Syamala Devi [24] focused on the complex interaction (e.g., multiple records) of agents by means of ontologies. Lavbič et al. [25] used ontologies (1) for providing a common understanding of a problem and (2) for communication between users and agents. Luncean and Becheru [26] proposed a message-based communication between agents by employing a message content ontology to capture the meaning of message. Ma et al. [22] reviewed various research works that made use of ontologies in MAS from the energy area. Rakib and Uddin Faruqui [27] employed ontology-driven rules to simply model real-world rules. Recently, Makris et al. [28] used ontologies for control and configuration purposes.

FCA is a powerful mathematical framework that underpins a large number of lattice-based approaches for organising, browsing, and querying data. Thus, FCA is suitable for decision-making. In [29], the authors devised a lattice-based tool, called ULYSSES, for visualising, browsing, and querying a lattice of concepts. The tool is used for information retrieval and allows domain experts to bound the search space by employing constraints based on their knowledge about the domain/goal. Ducrou et al. [30] presented the D-SHIFT web application for exploring a database concerning mobile phones and assisting a customer in purchasing a new phone. Relying on the lattice of concepts built by using FCA, the customer makes decisions and compromises in order to opt for the phone whose features satisfy the search criteria. Braud et al. [31] used FCA to (1) reveal and visualize the intrinsic structure of hydro-ecological data about water bodies in the Alsace plain and (2) assess a new water area. Thus, a lattice of concepts was derived for clustering water areas with similar biological and physicochemical parameters. The concepts were interpreted and annotated by domain experts. Then, a new water area was assessed by querying the lattice. Codocedo et al. [32] presented an FCA-based approach for handling querying over a concept lattice of documents and annotations. The authors focused on non-matching documents, i.e., documents that are semantically relevant to the user query but do not satisfy the search criteria. To this end, the initial query is modified; then, "cousin concepts" are identified and ranked according to a similarity metric to present retrieved documents. Anantaram et al. [33] proposed a deep learning pipeline that supports natural language queries over relational databases. Relying on FCA, the results of the query are converted into a lattice of concepts that is used to predict the category of a new object.

In addition, there have been several works from the MAS area that have made use of FCA or its extensions. Gajdoš and Radecký [34] applied triadic FCA, i.e., an FCA extension, to analyse the behaviours of agents employed by a traffic simulation example. Specifically, agents are clustered according to their features in order to, e.g., highlight similar behaviours of agents or predict agent behaviours. Zaki and Gammoudi [35] used AOC-poset and FCA to develop a decentralised and scalable method for enabling autonomous agents to efficiently allocate tasks in a massive MAS.

A simple MAS for heifer management was developed as a proof of concept [36]. Thangaraj et al. [37] modelled am MAS platform for DFs. In short, this platform automatically integrates data from various tools (about, e.g., pasture and nutrients) and thus allows domain experts to perform a variety of data analyses to make inferences. However, Thangaraj et al. did not focus on the outsourcing part; there was no management of the negotiation.

In the present article, we use FCA to cluster DFs that share common attributes and thus to obtain a search space of potential partners (i.e., DFs). Additionally, we assess the DFs by annotating the concepts with a satisfaction degree.

Compared to previous works, this paper tackles the challenges of semantic interoperability by proposing an intelligent business model for a particular field (i.e., DFs). In addition, the proposed collaborative MAS automates the selection of suitable partners, as well as the management of similar proposals. As a result, in this research, we put stress on semantic interoperability by using an ontology shared by all involved negotiation agents.

4. Technical Instrumentation

FCA ([17]) yields a hierarchy of conceptual abstractions, referred to as a lattice of concepts, that are used to analyse objects. FCA encodes binary data into a formal context. A formal context K is a 3-tuple $(G,\ M,\ I)$, where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is a binary relation that specifies which objects have which attributes.

For instance, Table 2 shows a formal context $K = (G,\ M,\ I)$. The rows are the objects $G = \{DF1,\ DF2,\ DF3,\ DF4,\ DF5,\ DF6\}$; the columns are the attributes $M = \{NF,\ GS,\ PM,\ UPM,\ NtIA,\ NtF\}$, where NF, GS, PM, UPM, NtIA, and NtF stand for *natural_forage*, *growth_stimulant*, *pasteurised_milk*, *unpasteurised_milk*, *nextTo_industrialAreas*, and *nextTo_Forests*, respectively; and a cross in a cell, identified by a pair, e.g., $(DF1,\ NF) \in I$, signifies that object $DF1 \in G$ has the attribute $NF\ \in M$.

Table 2. The formal context K built from the raw data given in Table 1. Each column name is abbreviated (i.e., NF, GS, PM, UPM, NtIA, and NtF stand for *natural_forage*, *growth_stimulant*, *pasteurised_milk*, *unpasteurised_milk*, *nextTo_industrialAreas*, and *nextTo_Forests*, respectively).

K	NF	GS	PM	UPM	NtIA	NtF
DF1	×		×		×	×
DF2	×	×	×			×
DF3		×		×	×	
DF4	×		×	×		×
DF5		×		×	×	
DF6	×		×	×	×	×

Two derivation operators, both denoted $'$, are defined for $X \subseteq G$ and $Y \subseteq G$ as follows:

$$' : P(G)\ \to P(M),\ X' = \{m \in M | \forall g \in X, (g,m) \in I\},$$

$$' : P(M)\ \to P(G),\ Y' = \{g \in G | \forall m \in Y, (g,m) \in I\}.$$

The $'$ operators define a Galois connection between $P(G)$ and $P(M)$. The set X' is the set of all attributes in M shared by the objects in X. Similarly, Y' is the set of all objects in G

that have the attributes in Y. The composition operators $''$ are closure operators since they are idempotent, monotonous, and extensive [17]. When $X = X''$ and $Y = Y''$, then X and Y are closed sets.

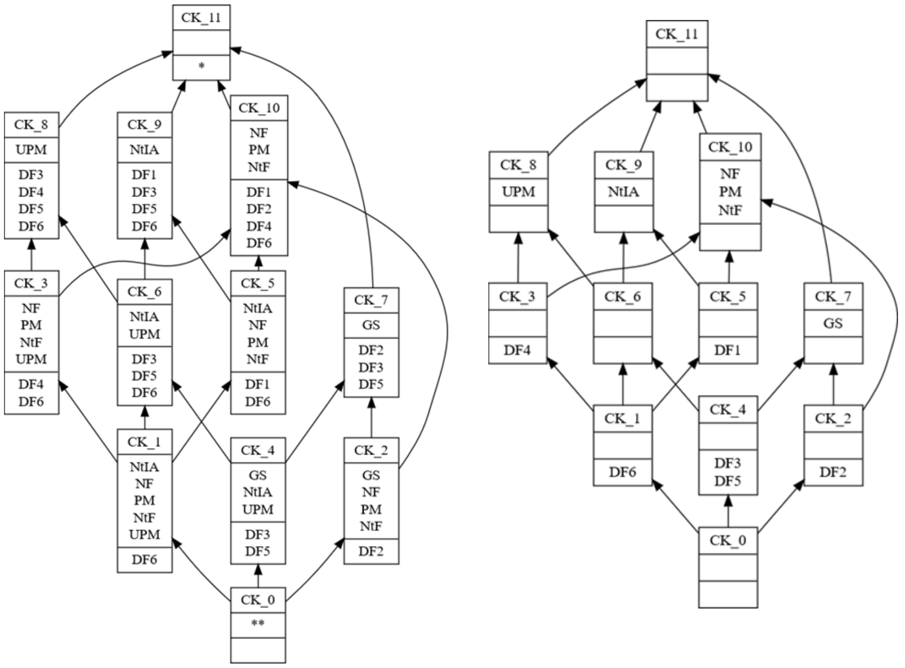
For example, relying on K given in Table 2, if $X_1 = \{DF3,\ DF5\}$, then $X_1{}' = \{GS,\ UPM,\ NtIA\}$, and $X_1{}'' = \{DF3,\ DF5\}$. Since $X_1 = X_1{}''$, X_1 is a closed set. In contrast, $X_2 = \{DF1,\ DF2\}$ is not closed because $X_2{}'' = \{DF1,\ DF2,\ DF4,\ DF6\} \neq X_2$.

A formal concept c derived from the context $K = (G,\ M,\ I)$ is a pair (X, Y), where $X \subseteq G$ and $Y \subseteq M$, s.t. $X' = Y$ and $Y' = X$. The set of objects X is called the extent of c, while the set of attributes Y is called the intent of c.

Let $c_1 = (X_1, Y_1)$ and $c_2 = (X_2, Y_2)$ be two concepts from C_K. The generalisation order $\preceq_K$ is defined by $c_1 \preceq_K\ c_2$ iff $X_1 \subseteq\ X_2 (\Leftrightarrow Y_2 \subseteq\ Y_1)$. In this case, c_1 is called a sub-concept of c_2, and c_2 is a super-concept of c_1. If $c_1 \prec_K\ c_2$ and there is no c_3 s.t. $c_1 \prec_K\ c_3 \prec_K\ c_2$, then c_1 is a lower neighbour of c_2 and c_2 is an upper neighbour of c_1 [38].

The concept lattice L_K of $K = (G,\ M,\ I)$ is the partial order $(C_K, \preceq_K)$. $\top(L_K)$ denotes the concept from C_K whose extent has all the objects in G, and $\perp(L_K)$ denotes the concept from C_K whose intent has all the attributes in M. Lattice L_K is represented by a Hasse diagram where the vertices are the concepts in C_K and the edges are defined by the relation $\prec_K$. The representation of L_K can be simplified as every attribute/object is top–down/bottom–up inherited. Therefore, an attribute/object is shown only in the highest/lowest concept where it appears.

For instance, in Figure 2a, CK_4 is a formal concept represented by a box structured from top to bottom as follows: concept name, intent $\{GS,\ NtIA,\ UPM\}$, and extent $\{DF3,\ DF5\}$. In addition, CK_2 $\preceq_K$ CK_7 (i.e., $\{DF2\} \subseteq \{DF2,\ DF3,\ DF5\}$) and CK_5 $\prec_K$ CK_10. Figure 2a depicts the L_K concept lattice built from K, while Figure 2b shows the simplified representation of the same L_K. Indeed, in Figure 2b, concept CK_6 has the extent $\{DF3,\ DF5,\ DF6\}$, where the objects $DF3$ and $DF5$ are inherited from CK_4, while object $DF6$ is inherited from CK_1. The intent of CK_6 is $\{UPM,\ NtIA\}$, where the UPM and $NtIA$ attributes are inherited from CK_8 and CK_9, respectively.



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

Figure 2. (**a**) The concept lattice L_K built from the formal context K (Figure 1); * is the set of all objects G; ** is the set of all attributes M; (**b**) the simplified version of L_K.

5. Collaborative Negotiation System

5.1. Negotiation Model

The following section defines the fundamental concepts underlying the proposed negotiation model.

A negotiation model is defined as a quintuple M = <T, P, N, R, O> where:

- T—the timestamp of the proposed negotiation system.
- P—the set of participants in the negotiation system; a participant can enter into one or more negotiations.
- N—the set of negotiations that happen within the negotiation system.
- R—the set of negotiation coordination policies that occur within the negotiation system. a coordination policy represents a set of rules that set up dependencies among ongoing negotiations.
- O—a common ontology that encompasses the definitions of the attributes that describe the negotiation object for a particular area (e.g., dairy products).

A negotiation is defined at a time point as a set of negotiation sequences. Let S = {si | i∈$\mathbb{N}$} indicates the set of negotiation sequences, such that $\forall si$, $sj \in$ S, i $\neq$ j implies $si \neq sj$. A negotiation sequence is $\in$ S such that $si \in$ N(t) is defined as a series of negotiation graphs that describe a negotiation N from the initial moment and up to a time point *t*.

A negotiation graph created at a given time point *t*, denoted G = (A, E) (where *A* is a set of nodes and *E* is a set of edges), is represented as an oriented graph; the nodes describe the negotiation proposals by specifying attributes and values at the time point *t*, and the edges indicate the proposal–counterproposal sequences.

Status is defined as a set of possible states for a negotiation. A state can take one of the following values: initiated, undefined, success, or failure. Initiated indicates the initial negotiation sequence, undefined indicates an ongoing negotiation sequence, while success and failure indicate the sequence in which an agreement has been reached or the negotiation has been ended with a denial, respectively.

The Issues concept is defined as a set of attributes with associated values that describe the negotiation offers.

Role is defined as a set of participant roles which can have one of the following values: initiator or guest. Initiator is the initial participant of the negotiation N, and guest represents the invited participant in the negotiation N.

5.2. Negotiation System Architecture

An MAS offers universal mechanisms to assist negotiations in a distributed environment through the interactions of deliberative agents [39].

The generic architecture of a negotiation system (Figure 3) has been described in previous papers [4,40]. The proposed approach considers that every participant (e.g., DF) has a negotiation infrastructure built as a single deliberative agent (noted with NegA) and a common negotiation middleware with all the participants. There, a human participant (i.e., Manager) can start a negotiation by providing his NegA with the negotiated task (i.e., Negotiation Object) and the way that the negotiation will be conducted (i.e., Negotiation Framework) to handle the negotiation with another NegA in the system.

A Negotiation Agent (NegA) supports its manager as follows: (1) at a global level (negotiations with dissimilar partners on different jobs) and (2) at a specific level (negotiation on the same job with dissimilar partners) [15]. To this end, a NegA coordinates itself with the negotiation agents of the other partners by means of the coordination and negotiation middleware.

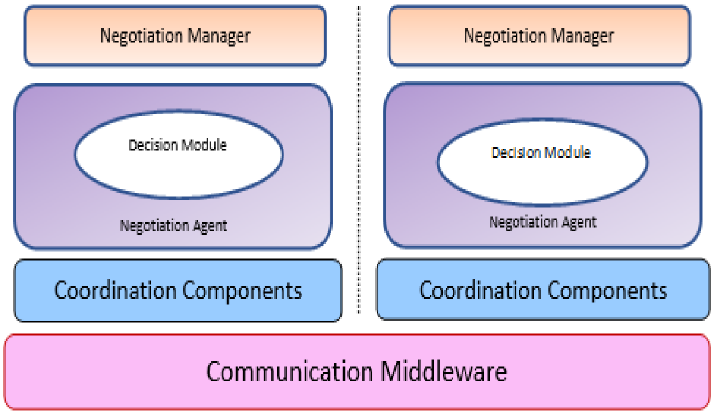


Figure 3. Negotiation architecture for each partner.

A negotiation comprises three key steps: initialisation, the refinement of the job under negotiation, and closing [41]. Initialisation allows one to define: (1) what must be negotiated (i.e., Negotiation Object) and (2) how (i.e., Negotiation Framework) [42]. Refinement allows participants to satisfy their constraints by exchanging proposals on the negotiation object [18]. A manager takes part in the initial definition of the negotiation frameworks and objects [43]. In addition, the manager, assisted by his/her NegA, make decisions by relying on the decision function from the “Reasoning” box (Figure 3). The NegA of a negotiation oversees a framework, one or more negotiation objects, and a negotiation status represented as a few graph structures. Furthermore, the manager may set out several global parameters, as follows: the maximum number of messages to be exchanged, maximum number of partners to be considered in the negotiation and involved in the contract, tactics, protocols for the NegA interactions with the manager and with other NegAs, and duration.

In direct contrast to [39], whose tactics manage the negotiation, here, tactics are used to define constraints on the negotiation process. For instance, a tactic may claim that a task must be outsourced as a block or be split up into several slots. Carrying out a tactic corresponds to triggering a combination of components that yields a coordinated modification of alternatives within the running negotiation [44]. A component manages a local view of the global negotiation; it converts decisions to modifications on the set of the visible alternatives on the task under negotiation by employing primitives of the negotiation middleware [45].

To cope with the complex types of negotiation scenarios, we put forward four dissimilar coordination components:

- Outsrc—the core component of a participant that initiates a negotiation for outsourcing tasks by exchanging proposals among possible reliable partners.
- Insrc—the key component of a guest partner that is invited in a negotiation for insourcing tasks.
- Block—the component guaranteeing that a task is thoroughly subcontracted by a single partner.
- Broker—a lattice-based component for automating the process of selection of reliable partners to begin the negotiation.

The aforementioned components are able to assess the received proposals, and, if these are valid, the components reply with new proposals drawn up based on their specific coordination constraints.

5.2.1. Coordination Negotiation Components

The coordination components have the role of managing the constraints among several concurrent negotiations. To solve the coordination issues, we divided the components into two distinct groups:

- Components operating in a closed environment: this group comprises the components whose images are constructed based on the ongoing negotiation. In this regard, the coordination constraints are managed according to the information coming from their negotiation graph (e.g., Outsrc, Insrc, and Block).
- Components operating in an open environment: similar to the first group, the images of these components are built on the basis of ongoing negotiations. The difference lies in the way the coordination constraints are managed. This is done based on the information available in the data structures related to other ongoing negotiations in the system (e.g., Broker).

In conclusion, the main difference between the two groups is the following: components in a closed environment manage the coordination constraints for a single negotiation, while components in an open environment manage the coordination of constraints among several negotiations that take place simultaneously.

Therefore, the main reason for introducing the Components layer is to coordinate all concurrent negotiations for a network participant. The two main functionalities of the Components layer are:

- Facilitating the communication between the Middleware and the Negotiation Agent.
- Facilitating the implementation of different negotiation cases depending on the chosen component (e.g., Block leads the negotiation so that the entire task is performed by a single partner).

5.2.2. Coordination Components in Closed Environment

The Outsrc component is the major component that initiates the negotiation. Therefore, starting from the object of negotiation, the initiation of an automatic negotiation is done by creating an instance of the Outsrc. Furthermore, this component will build the negotiation graph according to the negotiation requirements (i.e., evaluation and creation of offers and counter-offers). Negotiation requirements are met by Outsrc by manipulating Xplore primitives [46].

In addition to these functionalities, the Outsrc must interpret and verify the negotiation constraints that are defined in the following two data structures: Negotiation Object and Negotiation Framework. The Negotiation Object provides information on the possible values of the attributes to be negotiated. Based on this information, Outsrc verifies whether the offers received refer to the attributes negotiated in the current negotiation and whether they are associated to the values of the specified range. For instance, assuming that the price attribute set in the Negotiation Object must be in the range of 50k < price < 100k, the Outsrc will stop the negotiation in the white nodes in which the offers are outside the specified range.

Moreover, Outsrc coordinates the transactional aspect of the negotiation at the middleware level. Its main role is to set an agreement between the component instances participating in the same negotiation.

Another function of Outsrc is to execute the negotiation tactic defined in the Negotiation Framework by connecting the different instances of the other components.

The Insrc component is the main component that manages the negotiation for a guest partner. While Outsrc has a complete image of the ongoing negotiations, Insrc has only the image of its own negotiation. In addition, at the beginning of the negotiation, this component has no information about what is being negotiated or about the constraints imposed by its own manager.

Thus, unlike Outsrc, which has information from the beginning about the constraints defined in the two data structures (i.e., Negotiation Object and Negotiation Framework), Insrc communicates with its own manager about the new aspects required in the negotiation. Therefore, depending on the attributes defined by the initiator of the negotiation, Insrc is able to progressively build data structures that describe the Manager's preferences regarding the object of the negotiation and the negotiation framework.

The main functionality of the Block component is to coordinate the negotiations in which the task must be entirely performed by a single partner. In this regard, the Block's major role is to mediate the negotiation between the initiating DF and all the other DFs that are invited to the current negotiation. The purpose of mediation is to establish a contract regarding the performance of the entire task by a single partner. In this respect, Block manages the constraint of not dividing the subcontracted task into distinct parts.

In order to exemplify the operation of the Block component, we consider a negotiation initiated by the DF1 dairy farm and three other guest dairy farms (DF2, DF3, and DF4). In this context, the negotiation will start by initializing the Outsrc component of DF1. In order to fulfil the constraint regarding the accomplishment of the subcontracted task by a single dairy farm, Outsrc will invite the Block component in the negotiation. Next, Block connects to the Insrc component of each dairy farm (DF2, DF3, and DF4) and will manage the bilateral negotiations with them at the same time.

The process of interaction among participants begins when all components are connected. During this process, the Negotiation Agent of each DF involved in the negotiation begins to generate and to exchange offers and counteroffers for the task-in-progress. The negotiation ends when DF1 reaches an agreement with one of the participants (e.g., DF2) on the set of attributes that describe the negotiated task. At the same time, the initiator DF1 ends the negotiation with DF3 and DF4, this coordination being provided by the Block component. Otherwise, the negotiation can be concluded without reaching an agreement (e.g., the deadline set for the negotiation has expired or the three participants are no longer interested in the negotiation).

5.2.3. Coordination Components in Open Environment

The objective of defining the coordination components in an open environment is to allow for the coordination of the constraints among several negotiations that take place in parallel. Consequently, the main functionality of the components is to manage the flow of information coming from outside the current negotiation and, in particular, the information coming from other negotiations in which the considered DF is simultaneously involved. Unlike the coordination components in closed environments, which manage the flow of information among dairy farms within the same negotiation, the coordination components in an open environment manage the flow of information within the same dairy farm but for multiple bilateral negotiations.

Considering that middleware does not provide communication mechanisms among the instances of the components involved in different negotiations and, at the same time, the coordination components in an open environment cannot invite other components in a negotiation, we offer these components the ability to connect to particular data structures. This mode of communication is known as the "blackboard system". In this regard, for each type of component in an open environment, we propose a data structure. Using these data structures, the components are able to read the information from the external environment and, in particular, the information on the other existing negotiations. Furthermore, they are also able to write in these data structures in order to update the information on the negotiation in which they are involved. The information stored in these structures is necessary to establish the relations of dependence among several negotiations.

An example of the necessary information is the identifiers of the participants involved in the negotiations for the coordination of the transport of the negotiated tasks. This information allows for the coordination of the transport procedure among the considered negotiations [47]. A key aspect of this approach is the possibility to specialize the components capable of coordinating this type of dependencies.

To validate this approach, we propose the Broker component that provides the mechanism for selecting possible partners based on the information stored in these data structures.

The functionality of the Broker component is to find reliable partners who can perform a certain type of task. Its image on the graph is limited to the data contained in the node in which it was connected, but the Broker component additionally has access to the database of alliance partners (i.e., which includes name, address, and description of the type of job they can perform, e.g., milk production).

In the described scenario (Section 2), the negotiation between DFs is a multi-bilateral and multi-attribute negotiation. We assume that the manager of the DF initiating the negotiation does not specify the possible partners and delegates this task to the infrastructure (i.e., to the Broker component). Thus, the partners will be able to be known only during the negotiation and not at its initiation.

The selection process is based on the description of the negotiated task and the tracking of the alliance partners' database. Thus, in Figure 4, the Outsrc component initiates the negotiation by asking the Broker component to find partners who can perform a certain type of task. Broker navigates a search space of partners (i.e., a lattice of concepts generated with FCA) and returns the suitable partners with their satisfaction degrees (defined by domain experts) to Outsrc. Then, Outsrc immediately notifies the NegA by storing the list of partners into an in-memory table. In the following section, we detail the core of the Broker component.

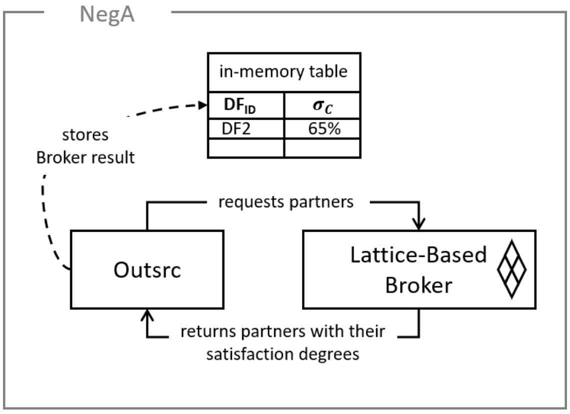


Figure 4. The interactions among the NegA, Outsrc, and Broker.

Here, we propose a lattice-based Broker component. To select the reliable partners of the negotiation task, the Broker component relies on FCA for visualising and exhibiting the intrinsic structure of the historical data about DFs. Precisely, FCA reveals a conceptual hierarchy (i.e., a lattice of concepts) that clusters DFs with common attributes (e.g., type of milk). For each generated formal concept c (i.e., cluster), domain experts are able to propose a profile (e.g., organic DF) and/or define a satisfaction degree σ_c (which measures the overall experience with that specific group of DFs).

Next, we present a manner to compute the satisfaction degree. Let $K = (G, M, I)$ be the formal context of the analysed DFs, where G is the set of DFs, M is the set of the characteristics of DFs, and $I \subseteq G \times M$ is a binary relation that specifies the characteristics for each DF. Firstly, domain experts categorise the characteristics of DFs as good or bad (see Table 3) according to their positive or negative impact on the quality of dairy products. Secondly, to discriminate between the "more or less" negative effect of bad characteristics, a penalty function $\surd : \mathcal{B} \rightarrow \mathcal{P}$—where $\mathcal{B} = \{x \in M | x\ belong\ to\ the\ bad\ category\}$ and $\mathcal{P} = \{p \in \mathbb{N}^* | \exists b \in \mathcal{B}\ \exists p\ a\ penalty\ defined\ by\ domain\ experts\}$—is used. Lastly, for a formal concept $c = (X, Y)$ generated from K, the satisfaction degree σ_c is computed by using Equation (1):

$$\sigma_c = \frac{|\{m \in Y | m\ belong\ to\ the\ good\ category\}|}{|Y|} \times 100 - \sum_{i=1}^{n} \surd(x_i) \tag{1}$$

where $n = |\mathcal{B}|$ and $x_i \in \mathcal{B}$.

Table 3. An example of categorising the DF characteristics.

Category	DF Characteristics	Penalty (p)
Good	Natural forage (NF)	–
	Pasteurised (PM)	–
	Forests (NtF)	–
Bad	Growth stimulant (GS)	10%
	Unpasteurised (UPM)	2%
	Industrial areas (NtIA)	7%

Equation (1) can be generalised for a family of penalised categories $\mathfrak{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_l\}$ as follows:

$$\sigma_c = \frac{|\{m \in Y | m \; belong \; to \; the \; unpenalised \; category\}|}{|Y|} \times 100 - \sum_{i=1}^{|\mathfrak{B}|} \sum_{j=1}^{n_i} \sqrt{}(x_j) \qquad (2)$$

where $n_i = |\mathcal{B}_i|$, $\mathcal{B}_i \in \mathfrak{B}$, and $x_j \in \mathcal{B}_i$. It is worthwhile to mention that the satisfaction degree may enhance the negotiation task since DFs can be easily ranked. For instance, a negotiation path with a DF can be ceased if the proposal is not consistent with its satisfaction degree, or the proposal with the highest satisfaction degree can be chosen among similar ones.

As a result, the Broker component is underpinned by the annotated lattice of concepts that represents the search space of potential DF partners. This search space is exploited by the Broker component in two ways, as follows:

- To retrieve the set of DFs that share a given set of attributes (denoted by Y). The query is the set of attributes Y; the lattice is browsed for a matching concept $c_1 = (X_1, Y_1)$ with $Y = Y_1$ (if it exists) or the most general concept $c_2 = (X_2, Y_2)$ with $Y \subset Y_2$. This can be done with various algorithms/tools (e.g., [30]) without modifying the queried lattice.
- To find the satisfaction degree of a given DF identifier (denoted by DF_{ID}) by looking for the most specific concept whose extent contains DF_{ID}. Thus, even if DF_{ID} may exist in many concepts of the lattice, only the concept that introduces DF_{ID} provides the accurate profile of the analysed DF.

For example, let us say that the Broker component has to pick two DFs whose key characteristics are animals fed with natural forage, produce pasteurised milk, and location next to forest. Relying on the search space of concepts L_K illustrated in Figure 2a, the Broker component finds an exact match, precisely concept CK_10. As a result, there are only four DFs that may attend the negotiation, namely DF1, DF2, DF4, and DF6. To choose only two DFs, the Broker component yet again makes use of the search space of partners. However, this time, the Broker component focuses on the satisfaction degree that annotates each most specific concept.

Figure 5 shows an excerpt of the L_K lattice built from the DF context depicted in Table 2. The most specific concepts that introduce DFs are annotated with their satisfaction degrees. In addition, a more user-friendly visualisation (i.e., pie charts) of the concept intents is proposed to engage managers.

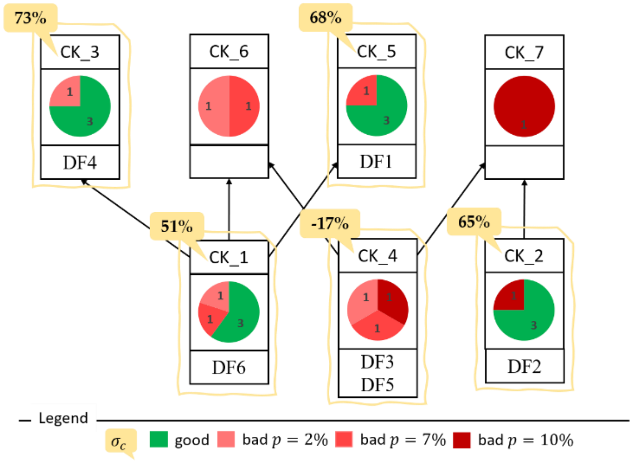


Figure 5. An excerpt of the simplified version of L_K (built from Table 1) where the most specific concepts are annotated with their satisfaction degrees σ_c. The intent relies on a more user-friendly visualisation (i.e., pie chart) to engage managers.

By relying on the simplified version of L_K (see Figure 5), domain experts can easily identify the concepts that introduce DFs, namely CK_1, CK_2, CK_3, CK_4, and CK_5. There are six analysed DFs but only five highlighted concepts (in yellow) because CK_4 clusters DF3 and DF5. These DFs share the same set of attributes (i.e., *growth_stimulant*, *unpasteurised_milk*, and *nextTo_industrialAreas*), which may have an unfavourable effect on the quality of dairy products; as a result:

$$\sigma_{CK\_4} = \frac{0}{3} \times 100 - (2\% + 7\% + 10\%) = -17\%.$$

The extent of CK_3 contains two DFs, namely DF4 and DF6 (see Figure 2a). However, only *DF*4 has $\sigma_{CK\_3} = 73\%$ since it is described by *natural_forage*, *pasteurised_milk*, *unpasteurised_milk*, and *nextTo_forests*. Indeed, the CK_1 extent contains *DF*6 and has the additional attribute *nextTo_industrialAreas*; this attribute may have an adverse impact on the satisfaction degree, the value of which decreases to $\sigma_{CK\_1} = 51\%$. Similarly, the extent of CK_5 contains DF1 and DF6, but only DF1 is introduced by this concept and has $\sigma_{CK\_5} = 68\%$. The extent of CK_2 concept introduces DF2 and $\sigma_{CK\_2} = 65\%$.

To wind our example up, based on the satisfaction degree the four DFs are ranked descending as follows: DF4, DF1, DF2, and DF6. Thus, the Broker component picks DF4 and DF1 that have the highest values.

It is worth mentioning that the L_K is built from scratch only once, after which it can easily be updated with new DFs. In short, a new DF (i.e., a new object) is added to the lattice of concepts by using an incremental algorithm (e.g., as explained in [31]). The new DF can be either added to an existing concept or added to a new concept, which has to be interpreted by domain experts. For instance, Figure 6 depicts the L_K lattice updated with *NEW_DF* that is described by *natural_forage*, *unpasteurised_milk*, and *nextTo_forests*. The CK_6 concept is the one which introduces *NEW_DF*. By analysing the highlighted path, the new DF is similar to some extent with two existing DFs, as follows: it has three out of four common attributes with DF4 and three out of five common attributes with DF6. As a result, domain experts may rely on the interpretation of DF4 to assess *NEW_DF*.

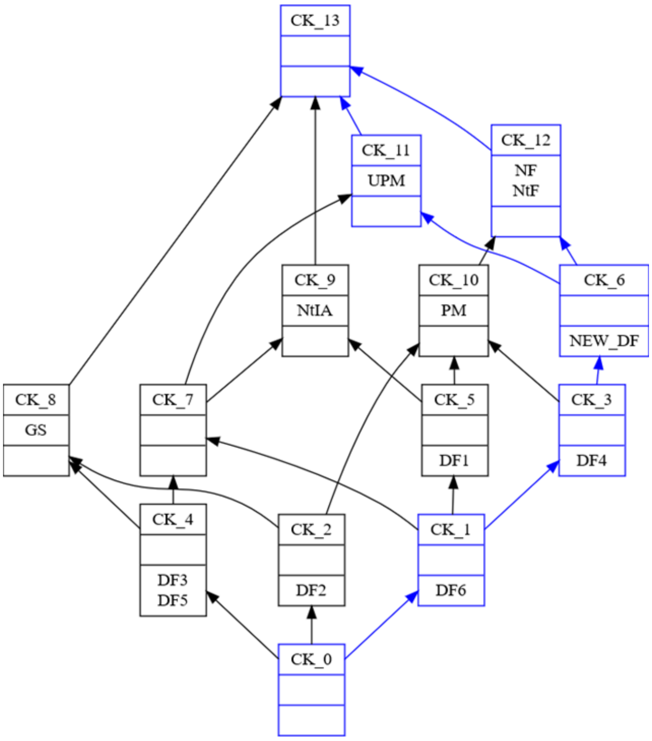


Figure 6. The simplified version of L_K updated with a new dairy farm *NEW_DF*.

5.3. Communication Process

The communication process is provided by the Middleware layer, which defines the generic mechanisms for broadcasting and synchronizing the offers elaborated during a negotiation process. This layer is an extension of the CLF (Coordination Language Facility) middleware [48] that is needed, in particular, to enrich the possibilities to support the various collaborative activities that take place within the alliance [49].

This extension is called Xplore. At the Xplore Middleware level, a negotiation process is represented as a bicoloured graph. The evolution of a negotiation, in terms of proposals and counterproposals, corresponds to the construction of a black and white graph. In this graph, the white nodes contain information about the attributes of the negotiated task, while black nodes represent decision nodes that may have multiple alternatives. White nodes are characterized by a parameter and a set of attributes with associated values. The parameter is the Negotiation Object that is described at a given moment by the negotiated attributes, depending on the context of the negotiation in the respective white node.

During a negotiation, several proposals and counterproposals can be exchanged, so there are several alternatives to continue the negotiation [50]. These alternatives are modelled by black nodes that are, consequently, nodes of exclusive choice among several white nodes. The notion of choice introduced by the black nodes imposes a restriction on the construction of the Xplore graph, namely that sub-graphs that have a common black root must not have any other node in common.

According to the architecture of the proposed negotiation system (see Figure 4), at the Middleware level, the negotiation is managed as a process of synchronization of several partial copies of a bicoloured graph. The Coordination Components layer facilitates the instantiation of one or more components for a current negotiation (e.g., Outsrc, Broker, and Block).

Each instance manages, depending on its functionalities, a local image of the current negotiation through a bicoloured graph corresponding to the respective agent. These instances interact with the Middleware layer through Xplore primitives to synchronize different bicolour graphs. For example, a negotiation, which involves a potential initiator (DF1) and three possible partners (DF2, DF4, and DF6), requires an instance of the Outsrc component that handles the negotiation for the initiator and three instances of the Insrc component for each possible partner.

According to the proposed approach wherein the negotiation process is distributed among the alliance participants, the middleware Xplore builds the negotiation graphs in the same manner for each of the partners. Hence, each negotiation participant makes decisions and acts on its own copy of the Xplore graph. Therefore, only the initiator (DF1) has a global view of the negotiation graph, whereas the guests (DF2, DF4, and DF6) have a partial view corresponding to the offers sent to each of them by the initiator.

In this respect, the Middleware layer must maintain the graphical image of the negotiations for each participant and synchronize this image with those of the partners involved.

To model this synchronization, the Middleware uses six operations (*connect*, *open*, *quit*, *assert*, *request*, and *ready*) to manipulate the graphs. Therefore, the partners communicate indirectly through the proposed operations called verbs. These verbs propagated by Middleware are used by each participant on their copy of the negotiation graph.

Subsequently, we explain each verb based on which the communication of the partners is made in the context of a negotiation in the field of dairy products.

- Connect(n,p): this informs a participant, identified by the parameter p, that is invited to a negotiation that has the root n. This identification is necessary to make a clear distinction among the different Xplore graphs and the tasks negotiated at the level of each partner involved in the same negotiation.

For example, we assume that the dairy farm DF1 wants to start a negotiation with two partners of the alliance (e.g., DF2 and DF3) in order to buy a quantity of 1000 kg of dairy products. By using the verb *connect* in two different nodes, participants DF2 and DF3 are introduced by DF1 into the current negotiation and, depending on their root white nodes, will have different images for the same negotiation (Figure 7).

- Open(n,n_1, … , n_i): this adds a new node n from parent nodes n_1, … , n_i (if any). The colour of the new created node must be the opposite of the colour of the parent nodes (i.e., the parent nodes must have the same colour).
- Assert(v, a): this communicates a negotiation offer regarding the value of the variable v about the aspect a.
- Request(v, a): this informs the partners that the initiator is waiting for an offer concerning the decision variable v about the aspect a.

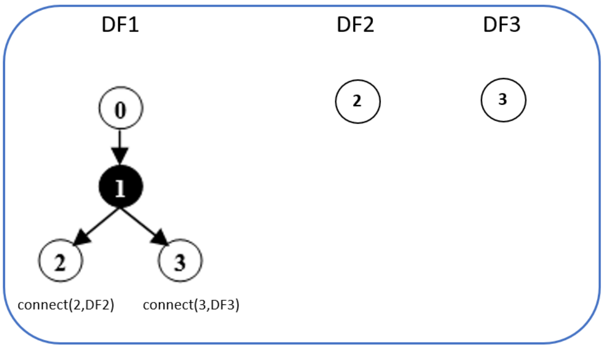


Figure 7. Connect verb.

Continuing the previous example, we suppose that DF1 wants to negotiate a task for a quantity of 1000 kg of dairy products and also wants to make the same proposal to the two guests, DF2 and DF3. Considering that node 0 is the root of the graph seen by DF1, the initiator will use the inheritance property of the graph to make the same proposal to both guests. Therefore, the initiator DF1 uses the verb *assert*(*quantity* = *1000*) to enter the attribute *quantity* with the value = 1000 in the root of the negotiation graph (i.e., node 0). Next, DF1 informs the other two participants that he expects a proposal for the attributes: price and fat; this announcement is made by the verb *request(price)*, *request(fat)*.

Subsequently, we assume that the guest DF2 wants to make two distinct proposals about the price of a quantity of 1000 kg of dairy products depending on their quality regarding the milk fat. Therefore, in the first stage, DF2 will open a black node 4 from the white node 2—*open(4,2)*—and then, starting from this black node, he will initiate the proposals by opening two white nodes—*open(5,4)* and *open(6,4)*. Furthermore, in these white nodes, DF2 will describe the proposals using the assert verb: *assert(price = 50, fat = 3)* and *assert(price = 60, fat = 3.5)*.

Similarly, using the same verbs, the guest DF3 will initiate one proposal: *open(7,3)* black node, *open(8,7)* white node, and *assert(price = 60, fat = 4)* (Figure 8).

- Ready(*n*)*:* this informs that a participant is satisfied with the information in the white node *n* and wants to accept the proposal.
- Quit(*n*)*:* this announces that a participant does not want to pursue the negotiation in the white node *n*.

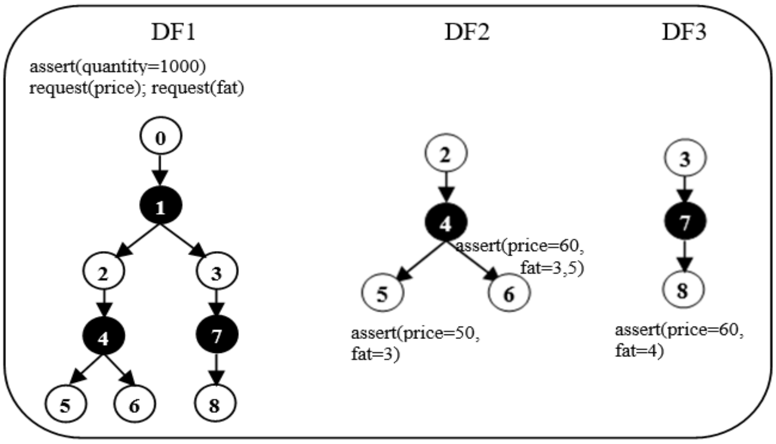


Figure 8. *Open*, *assert* and *request* verbs.

In the previous example, we suppose that the initiator DF1 is content with the proposal made by DF3 in node 8. As a result, DF1 stops the negotiation in nodes 5 and 6, *quit(5)* and *quit(6)*, respectively, and accepts the proposal in node 8, *ready(8)* (Figure 9).

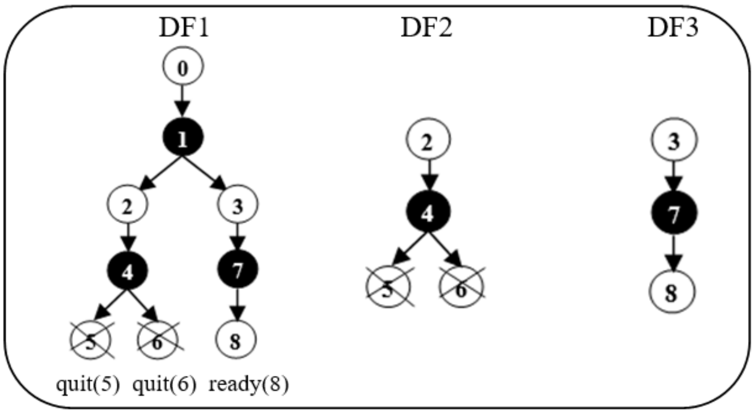


Figure 9. *Quit* and *ready* verbs.

6. Negotiation Scenario

We assume that a manager of DF1 receives a dairy product demand from a supermarket. For instance, the supermarket manager wants to buy two tons of high-quality dairy products from DF1. The DF1 manager cannot perform this demand alone when considering the availability of its dairy product resources. In this case, he/she decides to collaborate with other alliance partners by outsourcing a part of a job (e.g., 1-ton dairy products). In order to deliver high-quality dairy products, the initiating manager is interested in DFs that have animals fed with natural forage, pasteurised milk, and located next to the forest. To this end, he/she decides to initiate negotiations with a maximum of three DFs within the alliance. The manager specifies only the number of negotiation partners, but the process of selecting them is carried out by the Broker component. According to the proposed negotiation behaviours, we assume that DF1 manager wants to establish a contract regarding the execution of the entire job (i.e., 1-ton dairy products) by a single participant. This task will be performed by the Block component.

In this context, the negotiation starts by the initialisation of the Outsrc component of the initiator manager of DF1. Next, the Outsrc asks the Broker component to find maximum three DF partners with the following characteristics: animals fed with natural forage, pasteurised milk, and location next to the forest. Relying on the search space of potential partners depicted in Figure 2a Broker finds the CK_10 concept, which clusters three DFs that meet the requirements, namely DF2, DF4, and DF6. Then, Outsrc invites the Block component of the initiator DF1 to the negotiation. Further, Block connects to each partner's Insrc component (i.e., DF2, DF4, and DF6) and coordinates the three parallel bilateral negotiations with them. As soon as all components are connected, the interaction process among the negotiation partners may begin.

The negotiation proposals and counterproposals are generated by the NegA of each partner based on the following negotiating attributes: quantity, price, delay, fat, and animal.

Some attributes are specified by the negotiation object (i.e., quantity, price, and delay), and others are introduced based on the common ontology from the dairy product area illustrated in Figure 1 (i.e., fat and animal).

Let us say that the initiator DF1 wants to buy a quantity of high-quality dairy products of one ton (i.e., quantity = 1000 kg) with a reservation price 100/unit distributed in a maximum time of 5 h (i.e., delay <= 5 h). To ensure the high quality of dairy products, the milk fat must be at least 3% (i.e., fat >= 3%). Additionally, the manager wants the milk to be collected from Ovine (i.e., animal = Sheep or animal = Goat).

DF2, DF4, and DF6 are interested in selling dairy products. DF2 wants to have a reservation price 60/unit, while DF4 and DF6 want to have reservation prices 63/unit and 64/unit, respectively.

Assuming that the NegAs of DF2, DF4, and DF6 have a simple proposal generation mechanism starting from a minimal cost estimate (e.g., 60/unit for DF2, 63/unit for DF4, and 64/unit for DF6), then it increments with a fixed step (e.g., the bid increment of 4 for DF2, 3 for DF4, and 2 for DF6) until the proposal is accepted or the time allocated for the negotiation expires.

Table 4 illustrates how the three negotiations are simultaneously managed by the initiator DF1 and Block component.

As it can be seen, Table 4 shows how the initiator DF1 negotiates with DF2 starting with node 2, DF4 starting with node 3, and DF6 starting with node 4. Figure 10 represents the negotiation graph for the initiating partner, DF1, while Figure 11 represents the negotiation graphs for the three guests, DF2, DF4, and DF6.

This example shows how three parallel negotiations can be managed by the proposed negotiation infrastructure. Therefore, at the middleware level, the negotiation interactions correspond to the creation in the negotiation graph of three white nodes (i.e., nodes 2, 3, and 4) managed by the Outsrc component of DF1 (Figure 10). Further, these interactions are propagated as the creation of a white node in each negotiation graph managed by the Insrc component of each guest, DF2, DF4, and DF6 (Figure 11). The three guests have their own (partial) copy of the complete negotiation graph but no information about each other in the negotiation process. The negotiation process is described below:

- The NegA of the initiator DF1 sends the negotiation proposal to NegAs of the three invited dairy farms. The negotiation proposal is described by the following attributes: (quantity = 1000, delay <= 5, Fat >= 3, animal = ovine).

- The NegAs of the three guests look in the common ontology for the attributes to be negotiated (Figure 1). For example, for the animal = ovine, they find two values sheep and goat. Once they have understood all the attributes to be negotiated, NegAs will consult their managers, and they will generate various counterproposals.
- Subsequently, these counter-proposals are communicated with the help of the Insrc components of the three guests (DF2, DF4, and DF6) by opening the nodes in their own negotiation graphs, as well as in the graph managed by the Outsrc component of the initiator DF1.
- The negotiation continues until the initiator DF1 decides to accept a negotiation offer made in a white node by one of the three partners.
- Given the coordinated constraint of the Block component, namely that the entire quantity of dairy products to be purchased from a single partner, the initiator NegA must choose a single negotiation offer.
- In the proposed scenario, the two partners, DF4 and DF6, propose the same offer in white nodes 10 and 13, respectively. In this case, the initiator NegA will retrieve the satisfaction degree of both dairy farms by querying its in-memory table; then, NegA will choose the partner with the highest satisfaction factor. Taking this into account, NegA will accept the offer made by the partner DF4 in the white node 10 since $\sigma_{CK\_3} = 73\%$, while DF6 has $\sigma_{CK\_1} = 51\%$. At the same time, the initiator NegA will reject all other proposals from the other white nodes.

Table 4. Negotiation process description.

<table>
<tr><th colspan="3">Negotiations between the Initiator DF1 and Three Guests DF2, DF4, and DF6</th></tr>
<tr><td colspan="3">Step1: DF1: Open node 0 (white); assert(Quantity = 1000, Delay <= 5, Fat >= 3, Animal = Ovine); request(Price); request(Delay); request(Fat); request(Animal)
Step2: DF1: Open node 1 (black)</td></tr>
<tr><th>Negotiation between DF1 and DF2</th><th>Negotiation between DF1 and DF4</th><th>Negotiation between DF1 and DF6</th></tr>
<tr><td>Step3: DF1: Open node 2 (white); connect (2, DF2, Block)
Step4: DF2: Open node 5 (black) from 2
Step5: DF2: Open node 6 (white) from 5; assert (Price = 60; Delay = 5; Fat = 3; Animal = Goat)
Step 6: DF2: Open node 7 (white) from 5; assert (Price = 64; Delay = 4; Fat = 3,5; Animal = Sheep)
Step7: DF1: Quit node 7
Step8: DF1: Quit node 6</td><td>Step3$'$: DF1: Open node 3 (white); connect (3, DF3, Block)
Step4$'$: DF3: Open node 8 (black) from 3
Step5$'$: DF3: Open node 9 (white) from 8; assert (Price = 63; Delay = 5; Fat = 3.5; Animal = Sheep)
Step6$'$: DF3: Open node 10 (white) from 8; assert (Price = 66; Delay = 3; Fat = 4; Animal = Goat)
Step7$'$: DF1: Quit node 9
Step8$'$: DF1, Block: Ready node 10</td><td>Step3$''$: DF1: Open node 4 (white); connect (4, DF4, Block)
Step4$''$: DF4: Open node 11 (black) from 4
Step5$''$: DF4: Open node 12 (white) from 11; assert (Price = 64; Delay = 4; Fat = 3.5; Animal = Sheep)
Step6$''$: DF4: Open node 13 (white) from 11; assert (Price = 66; Delay = 3; Fat = 4; Animal = Goat)
Step7$''$: DF1: Quit node 12
Step8$''$: DF1: Quit node 13</td></tr>
</table>

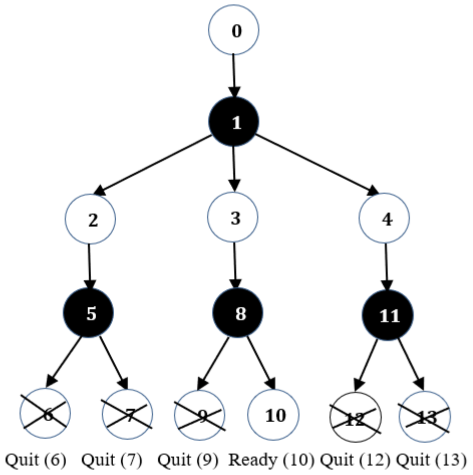


Figure 10. Negotiation graph for the initiator DF1.

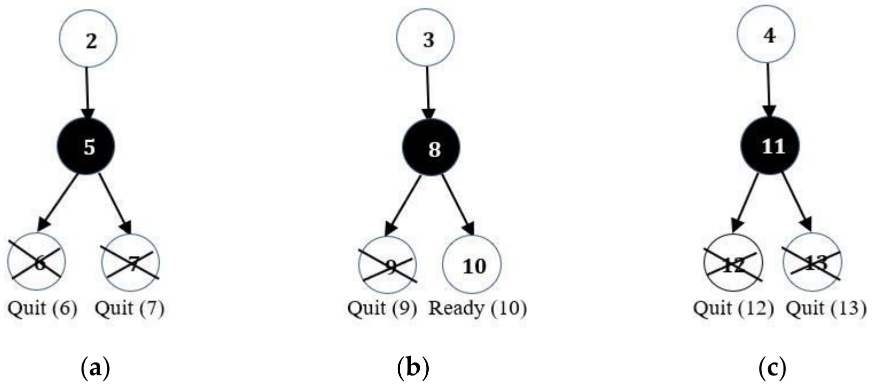


Figure 11. Negotiation graphs for the three guests: (**a**) DF2, (**b**) DF4, and (**c**) DF6.

7. Discussion

Many research studies have emphasized the importance of automated negotiation as a key role in dynamic trading in e-commerce.

In this context, several papers have addressed the issues of selected negotiation strategy and specific negotiation protocol [51–56].

For example, based on previous negotiations, Fujita [54] introduced an automated negotiation agent that could predict the strategies of the partners.

Likewise, Caillere et al. [56] tackled the problem of coordination of multi-party, multi-issue negotiations and proposed a protocol and rules that support agents to manage their interactions and reach an agreement.

However, many of these studies did not consider an important feature of human behaviour: the ability to express preferences in the negotiation process. In addition, other approaches assume that the agent is able to provide information about the history of previous negotiations.

Regarding the design of the negotiation environment, there are two main directions: (i) automated negotiations in which software agents replace human users in the negotiation process and (ii) semi-automated negotiations in which software agents assist and support human negotiators in the course of the negotiations or directly negotiate with a human user (i.e., agent-to-human interaction). Concerning the first direction, Alsrheed et al. [57] proposed a conceptual framework for achieving automated service level agreements (SLA) negotiation in which intelligent agents represent cloud providers and consumers. However, the negotiation coordination approach proposed herein is based on a one-to-one protocol in which an agent makes a proposal and the other agent can either accept or reject it without being able to respond with a counter-proposal.

Likewise, a recent study by Deochake and Mukhopadhyay [58] proposed an automated negotiation system for hybrid cloud computing that allows for concurrent negotiation transactions with multiple opponent agents. However, this approach has only one owner of the IT infrastructure looking to scale its infrastructure into the cloud. In addition, the paper focused on a single negotiation attribute (i.e., cost value) for each agreed issue and a time threshold.

A second direction has been found in many research studies [55,57–60]. However, there have been few studies that apply automated negotiations to real-world problems.

For instance, Park et al. [60] addressed the issue of multiple equivalent simultaneous offers for e-commerce transactions by proposing an efficient offer strategy model to achieve a win–win negotiation for an automated negotiation agent. This approach exhibits three attributes (issues) of the negotiation task over which agreement must be reached (e.g., unit price, warranty, and delivery date).

The main difference of the automated business model presented in this paper is in the proposed solution's ability to coordinate multiple concurrent negotiations.

Hence, we proposed a solution that splits the coordination issues into two groups depending on the complexity of negotiations (i.e., components in open and closed environment). Based on this distributed approach, we developed a parallel negotiation coordination model distributed over several coordination components.

The proposed solution allows one to cope with a scalable environment and easily decrease/increase the complexity of coordination by removing/adding independent coordination components.

Moreover, the division of the coordination model into independent coordination components makes it very flexible, thus allowing it to be used for simple interoperability negotiations (i.e., bilateral negotiation on simple attributes and with limited number of interactions) up to very complex interoperability problems (i.e., multiple participants, complex attributes, and dependent or nested negotiations). For example, the proposed scenario describes a complex negotiation consisting of three parallel bilateral negotiations carried out for a negotiation task described by five attributes specified by the negotiation object and by the common ontology.

Furthermore, compared to the negotiation models described above in which coordination issues are managed at the protocol level or by complexifying the utility functions, our approach defines and handles the synchronization negotiation mechanisms at the middleware level.

The advantage of the proposed coordination solution is the utility function that is transparent for the communication middleware and that is common to all parties involved in negotiation. Consequently, this generic solution can be integrated in any deliberative negotiation system or directly used as a support in an agent-to-human interaction.

The main idea of this paper is to propose an automated business model that helps small and medium-sized DFs to address the challenges of a competitive market. In this respect, the key solution to meet market requirements in a short time is based on automatic negotiation with other DFs in a collaborative virtual environment. Moreover, the partners involved in the negotiation can reach a common agreement through a unanimous decision that will ensure the interoperability of the system and will lead to a seamless collaboration across time.

Usually, a DF is autonomous and is in competition with others, but in order to consolidate its position on the market and to honour a challenging contract with minimal work, a manager may decide to collaborate with the alliance's partners to outsource parts of its work.

In order to maintain the autonomy of each participant within the alliance, the authors previously proposed a decentralised and flexible negotiation solution based on a cloud infrastructure that combines diverse technologies, such as multi-agent systems in dealing with negotiation interactions issues and middleware-level coordination facilities for all aspects related to coordination of multiple parallel negotiations.

Compared to previous works, this paper tackles the challenges of semantic interoperability by proposing an intelligent business model where a DF ontology is shared by all involved NegAs. Moreover, the proposed business model addresses the issue of the automatic selection of trusted partners by proposing a lattice-based FCA approach. The management of similar proposals is also considered in this paper. With reference to this, a satisfaction degree is defined for each possible partner.

The architectural design of the negotiation system of each DF follows the proposed distributed approach of dividing the negotiating process into three independent processes (i.e., communication, coordination, and decision-making process). Hence, generic mechanisms at the middleware level ensure synchronization of communication among agents using the Xplore protocol.

The coordination process is carried out by the proposed negotiation components, which can be used in real case negotiations. For instance, the presented scenario used the Outsrc component to coordinate the negotiations for the initiator in order to outsource a part of the work, Insrc components to coordinate the negotiations for the guests in order to insource the respective part of the work, the Broker component to automatically find reliable partners based on the satisfaction degree, and the Block component to ensure the coordination of negotiations based on the constraint established by the initiating manager (i.e., the entire work must be performed by a single partner).

The decision-making process is provided by the NegA, which helps the human user in dealing with all aspects regarding the establishment of the negotiation strategy, the evaluation and generation of proposals, and the protocol for exchanging proposals with other agents.

Limitations and Future Work

Several limitations need to be highlighted as perspectives for future work.

Regarding the future research, a new direction to optimise the proposal-generating algorithm and to reduce the negotiation time is to incorporate a learning mechanism. With the current proposal, agents are negotiating on behalf of the owners without having knowledge of the opponents' preferences and only limited information on their constraints. Providing a learning mechanism to estimate the opponent's preferences and constraints can reduce the negotiation space and allow agents to get to an optimal proposal faster.

By adding a learning mechanism and a proposal generation, our agent will become more autonomous, which will challenge our design statement aiming to have a mixed control between the human and the Negotiation Agent.

In this line of research, we need to provide functionalities that should increase the trust and adoption to lambda users of the proposed negotiation platform. Transparency in terms of agent's assumptions and reasoning, predictable impact on the opponent user, and allowing to share user reputation scores are possible directions of research in order to increase trust in negotiation agents.

Another research perspective is related to the description and implementation of other negotiation tactics specific to various business activities. More advanced studies will be further integrated into the negotiation process objectives, not only to find interoperability solutions but also to estimate the performance value and stability in time of each solution.

Furthermore, the authors aim to address the problem of ontology mapping and to find solutions to enable agents with different backgrounds to adjust themselves before collaborating with each other.

Author Contributions: Conceptualization, A.C. and C.N.; methodology, A.C., C.N., C.C., R.J.-G. and B.B.; validation, A.C., C.N., C.C., R.J.-G. and B.B.; formal analysis, Adina Cretan, C.N., C.C., R.J.-G. and B.B.; investigation, A.C., C.N., C.C., R.J.-G. and B.B.; writing—original draft preparation, A.C., C.N., C.C., R.J.-G. and B.B.; writing—review and editing, A.C., C.N., C.C., R.J.-G. and B.B.; supervision, R.J.-G. and B.B.; project administration, A.C., C.N., C.C., R.J.-G. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors wish to acknowledge and thank the support of the Portuguese Fundação para a Ciência e a Tecnologia through the project UIDB/04466/2020 to this research.

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Gherghina, Ș.C.; Botezatu, M.A.; Hossuzu, A.; Simionescu, L.N. Small and Medium-Sized Enterprises (SMEs): The Engine of Economic Growth through Investments and Innovation. *Sustainability* **2020**, *12*, 347. [CrossRef]
2. Johnson, M.P.; Schaltegger, S. Two decades of sustainability management tools for SMEs: How far have we come? *J. Small Bus. Manag.* **2016**, *54*, 481–505. [CrossRef]
3. López-Pérez, M.E.; Melero-Polo, I.; Vázquez-Carrasco, R.; Cambra-Fierro, J. Sustainability and Business Outcomes in the Context of SMEs: Comparing Family Firms vs. Non-Family Firms. *Sustainability* **2018**, *10*, 4080. [CrossRef]
4. Povolná, L. Innovation Strategy in Small and Medium Sized Enterprises (SMEs) in the Context of Growth and Recession Indicators. *J. Open Innov. Technol. Mark. Complex.* **2019**, *5*, 32. [CrossRef]
5. Gomes, G.; Wojahn, R.M. Organizational learning capability, innovation and performance: Study in small and medium-sized enterprises (SMES). *Rev. Adm.* **2017**, *52*, 163–175. [CrossRef]

6. Yadav, N.; Gupta, K.; Rani, L.; Rawat, D. Drivers of sustainability practices and SMEs: A systematic literature review. *Eur. J. Sustain. Dev.* **2018**, *7*, 531–544. [CrossRef]
7. Jelačić, D.; Pirc Barčić, A.; Oblak, L.; Motik, D.; Grošelj, P.; Jošt, M. Sustainable Production Management Model for Small and Medium Enterprises in Some South-Central EU Countries. *Sustainability* **2021**, *13*, 6220. [CrossRef]
8. Tsvetkova, D.; Bengtsson, E.; Durst, S. Maintaining Sustainable Practices in SMEs: Insights from Sweden. *Sustainability* **2020**, *12*, 10242. [CrossRef]
9. Lin, Y.-C.; Chen, C.-L.; Chao, C.-F.; Chen, W.-H.; Pandia, H. The Study of Evaluation Index of Growth Evaluation of Science and Technological Innovation Micro-Enterprises. *Sustainability* **2020**, *12*, 6233. [CrossRef]
10. Stone, P.; Veloso, M. Multiagent Systems: A Survey from a Machine Learning Perspective. *Auton. Robot.* **2000**, *8*, 345–383. [CrossRef]
11. Kumari, S.; Singh, A.; Mishra, N.; Garza-Reyes, J.A. A multi-agent architecture for outsourcing SMEs manufacturing supply chain. *Robot. Comput.-Integr. Manuf.* **2015**, *36*, 36–44. [CrossRef]
12. Ghobakhloo, M.; Hong, T.S.; Sabouri, M.S.; Zulkifli, N. Strategies for Successful Information Technology Adoption in Small and Medium-sized Enterprises. *Information* **2012**, *3*, 36–67. [CrossRef]
13. Vrchota, J.; Řehoř, P. Project management and innovation in the manufacturing industry in Czech Republic. *Procedia Comput. Sci.* **2019**, *164*, 457–462. [CrossRef]
14. Rizos, V.; Behrens, A.; Van der Gaast, W.; Hofman, E.; Ioannou, A.; Kafyeke, T.; Flamos, A.; Rinaldi, R.; Papadelis, S.; Hirschnitz-Garbers, M.; et al. Implementation of Circular Economy Business Models by Small and Medium-Sized Enterprises (SMEs): Barriers and Enablers. *Sustainability* **2016**, *8*, 1212. [CrossRef]
15. Cretan, A.; Coutinho, C.; Bratu, B.; Jardim-Goncalves, R. NEGOSEIO: A Framework for Negotiations toward Sustainable Enterprise Interoperability. *IFAC J. Annu. Rev. Control* **2012**, *36*, 291–299. [CrossRef]
16. Jardim-Goncalves, R.; Cretan, A.; Coutinho, C.; Dutra, M.; Ghodous, P. *Ontology Enriched Framework for Cloud-based Enterprise Interoperability, Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment, Vols*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1155–1166. [CrossRef]
17. Ganter, B.; Wille, R. *Formal Concept Analysis: Mathematical Foundations*; Springer: Berlin/Heidelberg, Germany, 1999.
18. Coutinho, C.; Cretan, A.; Jardim-Goncalves, R. Cloud-based negotiation for sustainable enterprise interoperability. In Proceedings of the 18th International ICE Conference on Engineering, Technology and Innovation, Munich, Germany, 18–20 June 2012; IEEE: Piscataway, NJ, USA, 2012.
19. Cretan, A.; Coutinho, C.; Bratu, B.; Jardim-Goncalves, R. A negotiation approach to support interoperability in a collaborative manufacturing environment. In Proceedings of the International Conference on Engineering, Technology and Innovation (ICE/ITMC), Funchal, Portugal, 27–29 June 2017.
20. Suarez-Romero, J.A.; Alonso-Betanzos, A.; Guijarro-Berdinas, B.; Duran-Sanles, C.A. A tool for agent communication in Mozart/Oz. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiegne, France, 19–22 September 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 706–710.
21. El-Ghamrawy, S.M.; El-Desouky, A.I. Distributed multi-agent communication system based on dynamic ontology mapping. *Int. J. Commun. Netw. Distrib. Syst.* **2013**, *10*, 1–24.
22. Ma, Z.; Schultz, J.M.; Christensen, K.; Værbak, M.; Demazeau, Y.; Jørgensen, N.B. The Application of Ontologies in Multi-Agent Systems in the Energy Sector: A Scoping Review. *Energies* **2019**, *12*, 3200. [CrossRef]
23. Malucelli, A.; Oliveira, E.D. *Ontology-Services to Facilitate Agents' Interoperability, in Intelligent Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 170–181.
24. Arora, M.; Devi, M.S. Ontology Based Agent Communication in Resource Allocation and Monitoring. *Int. J. Comput. Sci. Issues* **2010**, *7*, 28–32.
25. Lavbič, D.; Vasilecas, O.; Rupnik, R. Ontology-based multi-agent system to support businessusers and management, TechNologIcal aNd ecoNomIc developmeNT oF ecoNomYBaltic. *J. Sustain.* **2010**, *16*, 327–347.
26. Luncean, L.; Becheru, A. Communication and interaction in a multi-agent system devisedfor transport brokering. In Proceedings of the 2015 Balkan Confer-ence on Informatics: Advances in ICT, Craiova, Romania, 2–4 September 2015; pp. 51–58.
27. Rakib, A.; Uddin Faruqui, R.; Rakib, A.; Faruqui, R.U. Model checking ontology-driven reasoning agents using strategy and abstraction. *Concurr. Comput. Pract. Exp.* **2019**, *33*, e5205. [CrossRef]
28. Makris, S.; Alexopoulos, K.; Michalos, G.; Sardelis, A. An Agent-Based System for Automated Configurationand Coordination of Robotic Operations in RealTime—A Case Study on a Car Floor Welding Process. *J. Manuf. Mater. Process.* **2020**, *4*, 1–22.
29. Carpineto, C.; Romano, G. *ULYSSES: A Lattice-Based Multiple Interaction Strategy Retrieval Interface, in Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 91–104.
30. Ducrou, J.; Wormuth, B.; Klund, P.W.E. D-SIFT: A Dynamic Simple Intuitive FCA Tool. In *Conceptual Structures: Common Semantics for Sharing Knowledge, Proceedings of the 13th International Conference on Conceptual Structures, ICCS, Proceedings, Kassel, Germany, 18–22 July 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 295–306.
31. Braud, A.; Nica, C.; Grac, C.; le Ber, F. A Lattice-based Query System for Assessing the Quality of Hydro-ecosystems. In Proceedings of the Eighth International Conference on Concept Lattices and Their Applications, Nancy, France, 17–20 October 2011; pp. 265–277.

32. Codocedo, V.; Lykourentzou, I.; Napoli, A. A semantic approach to concept lattice-based information retrieval. *Ann. Math. Artif. Intell.* **2014**, *72*, 169–195. [CrossRef]
33. Anantaram, C.; Rastogi, M.; Rawat, M.; Saini, P. Enabling Natural Language Analytics Overrelational Data Using Formal Concept Analysis. Available online: http://ceur-ws.org/Vol-2529/paper1.pdf (accessed on 9 June 2021).
34. Gajdoš, P.; Radecký, M. Searching Common Pattern in Agents Behaviors with Usage of FCA. In Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, Italy, 30 November–2 December 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1190–1195.
35. Zaki, B.; Gammoudi, M.M. *Hierarchical Organization of Agents Based on Galois Sub-Hierarchy for Complex Tasks Allocation in Massive MAS, in KES-AMSTA 2009: Agent and Multi-Agent Systems: Technologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 460–470.
36. Parrott, L.; Lacroix, R.; Wade, K.M. Design considerations for the implementation of multi-agent systems in the dairy industry. *Comput. Electron. Agric.* **2003**, *38*, 79–98. [CrossRef]
37. Thangaraj, A.; Patricia, A.; Samarasinghe, S. Modelling a multi agent system for dairy farms for integrated decision making. In Proceedings of the 22nd International Congress on Modelling and Simulation, Hobart, Tasmania, Australia, 3–8 December 2017; pp. 529–535.
38. Roth, C.; Obiedkov, S.; Kourie, D.G. On Succint Representation of Knowledge Community Taxonomies with Formal Concept Analysis. *Int. J. Found. Comput. Sci.* **2008**, *19*, 383–404. [CrossRef]
39. Faratin, P. Automated Service Negotiation between Autonomous Computational Agent. Ph.D. Thesis, Department of Electronic Engineering Queen Mary & West-field College, London, UK, 2000.
40. Coutinho, C.; Cretan, A.; da Silva, C.F.; Ghodous, P.; Jardim-Goncalves, R.G. Service-based Negotiation for Advanced Collaboration in Enterprise Networks. *J. Intell. Manuf.* **2014**, *27*, 1. [CrossRef]
41. Hu, J.; Deng, L. An Association Rule-Based Bilateral Multi-Issue Negotiation Model. In Proceedings of the 2011 Fourth International Symposium on Computational Intelligence and Design, Hangzhou, China, 28–30 October 2011.
42. Luo, X.; Miao, C.; Jennings, N.R.; He, M.; Shen, Z.; Zhang, M. Kemnad: A knowledge engineering methodology for negotiating agent development. *Comput. Intell.* **2012**, *28*, 51–105. [CrossRef]
43. Jardim-Goncalves, R.; Coutinho, C.; Cretan, A.; de Silva, C.F.; Ghodous, P. Collaborative negotiation for ontology-driven enterprise businesses. *Comput. Ind.* **2014**, *65*, 1232–1241. [CrossRef]
44. Oliveira, A.I.; Camarinha-Matos, L.M. Electronic Negotiation Support Environment in Collaborative Networks. *Adv. Inf. Commun. Technol.* **2012**, *372*, 21–32.
45. Sycara, K.; Dai, T. *Agent Reasoning in Negotiation, in Advances in Group Decision and Negotiation*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 437–451.
46. Cretan, A.; Coutinho, C.; Bratu, B.; Jardim-Goncalves, R. A negotiation cloud-based solution to support interoperability among interconnected autonomous microgrids. In Proceedings of the 9th IEEE-TEMS International Conference on Intelligent Systems, Funchal, Portugal, 25–27 September 2018.
47. Jardim-Goncalves, R.; Cretan, A.; Coutinho, C.; Agostinho, C.; Bratu, B. Coordinating negotiations in data-intensive collaborative working environments using an agent-based model-driven platform. *Enterp. Inf. Syst.* **2018**, *12*, 1100–1128. [CrossRef]
48. Andreoli, J.-M.; Arregui, D.; Pacull, F.; Riviere, M.; Vion-Dury, J.; Williamowski, J. CLF/Mekano: A Framework for Building Virtual-Enterprise Applications. In Proceedings of the 3rd International Conference on Enterprise Distributed Object Computing, Mannheim, Germany, 27–30 September 1999.
49. Andreoli, J.-M.; Castellani, S. Towards a Flexible Middleware Negotiation Facility for Distributed Components. In Proceedings of the "E-Negotiation" DEXA, Munich, Germany, 1 June 2001.
50. Georgantas, N.; Rahaman, M.A.; Ameziani, H.; Pathak, A.; Issarny, V. A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems. In *Advances in Grid and Pervasive Computing*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 221–232.
51. Fernandez, S.; Ito, T. An Ontology to support automated negotiation. *arXiv* **2017**, arXiv:1710.10433.
52. Cao, M. Dynamic Time-Dependent Strategy Model for Predicting Humans Offer in E-Commerce Negotiation. *Open J. Soc. Sci.* **2016**, *4*, 64–69.
53. Nazari, Z. Automated Negotiation with Humans. Ph.D. Thesis, University of Southern California, Los Angeles, CA, USA, 2017.
54. Fujita, K. Automated Negotiating Agent with Strategy Adaptation for Multi-Times Negotiations. Recent Advances in Agent-Based Complex Automated Negotiation. *Stud. Comput. Intell.* **2016**, *638*, 21–37. [CrossRef]
55. Shojaiemehr, B.; Rahmani, A.; Qader, N. A three-phase process for SLA negotiation of composite cloud services. *Comput. Stand. Interfaces.* **2019**, *64*, 85–95. [CrossRef]
56. Caillere, R.; Arib, S.; Aknine, S.; Berdier, C. A Multiagent Multilateral Negotiation Protocol for Joint Decision-Making. Recent Advances in Agent-Based Complex Automated Negotiation. *Stud. Comput. Intell.* **2016**, *638*, 71–88. [CrossRef]
57. Alsrheed, F.; El Rhalibi, A.; Randles, M.; Merabti, M. Intelligent Agents for Automated Cloud Computing Negotiation. In Proceedings of the 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 14–16 April 2014.

58. Deochake, S.; Mukhopadhyay, D. An Agent-Based Cloud Service Negotiation in Hybrid Cloud Computing. In *ICT Systems and Sustainability. Advances in Intelligent Systems and Computing*; Tuba, M., Akashe, S., Joshi, A., Eds.; Springer: Singapore, 2021; Volume 1270. [CrossRef]
59. Hussain, I.; Khan, M.A.; Baqueri, S.F.A.; Shah, S.A.R.; Bashir, M.K.; Khan, M.M.; Khan, I.A. An Organizational-Based Model and Agent-Based Simulation for Co-Traveling at an Aggregate Level. *Appl. Sci.* **2017**, *7*, 1221. [CrossRef]
60. Park, J.; Rahman, H.A.; Suh, J.; Hussin, H. A Study of Integrative Bargaining Model with Argumentation-Based Negotiation. *Sustainability* **2019**, *11*, 6832. [CrossRef]