Hindawi
Scientific Programming
Volume 2020, Article ID 8898719, 12 pages
https://doi.org/10.1155/2020/8898719

Research Article

Cooperative Business Intelligence Model Using a Multiagent Platform

Mohamed O. Khozium [1] and Norah S. Farooqi [2]

[1] *Faculty of Public Health and Health Informatics, HITM Department, ICRS Consultant, Umm Al-Qura University, Mecca, Saudi Arabia*
[2] *College of Computer and Information Systems, Umm Al-Qura University, Mecca, Saudi Arabia*

Correspondence should be addressed to Mohamed O. Khozium; dr.khozium@gmail.com

Received 11 August 2020; Revised 11 October 2020; Accepted 31 October 2020; Published 11 November 2020

Academic Editor: Jiwei Huang

Copyright © 2020 Mohamed O. Khozium and Norah S. Farooqi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modern companies wish to utilize business intelligence (BI) to track and analyze their courses of action. Many BI applications serve this purpose at many levels, starting from documenting and charting and ending with analytics and decision support systems, which are considered a sufficient complement to consultancy and management resources. However, the contemporaneous BI software is missing two functionalities. First, although nearly all applications of the same genre use almost identical concepts, there is no unified application programming interface (API) to enable interaction. The second problem is a consequence of the first issue. Without a unified API, BI applications cannot be integrated, eliminating any possibility of establishing universal platforms for BI distributed services. Lacking these two functionalities makes developers reinvent the wheel with each new implementation. To solve these problems, we propose a platform running a multiagent business intelligence system. This system empowers the available BI resources to serve a larger segment of the BI end-user applications cooperatively. To build this system, we propose a unified model that enables distributive agent-based tasking and cooperative interaction. This allows researchers to cooperate in spreading the multiagent platform's functionality and helps them proceed toward more detailed analysis considering agents' construction. Moreover, it will enable BI service providers to cooperatively implement new applications and develop better solutions while maintaining a functional end-user program.

1. Introduction

The impact of a comprehensive feasibility study and smart administration upon business success is undeniable. There are many ways to measure the success of an organization. The financial scalar is on top of the list. This measurement indicates that success implies efficient strategies and favorable decisions and actions [1]. Consultants and experts usually provide the most feasible proposals based on the business's situation and the knowledge gained by them due to their research and experience.

Business intelligence (BI) is a demanding trend requirement for all businesses. Running a business today without using BI techniques is like walking blind, trying to reach a destination without being aware of the road's obstacles. Usually, consultants and experts provide services to help founders conduct an analytical feasibility study on their project, including alternative procedures and any affecting factors. Proper management for the project would ensure that the feasibility study would lead to the real world's best potential outcomes [2–4]. There are many digital tools to serve and aid consultants, experts, and managers in analyzing and determining the best actions and decisions, starting from a simple calculator, proceeding to digital planning software, and culminating with the most modern BI and decision support tools [5].

Although consultancy and management strategies have proven their feasibility in almost every business, the majority of startups and established businesses do not employ them due to two interacting factors: cost and availability. There are not enough consultants and experts to serve all industries, so the price determines the buyers [3]. There is no guarantee on the results of the advice provided by administrative consultants; there is always a measure of risk. However, that risk can grow so high that it is almost inevitable. The common factors in reaching this higher threshold are comprehensive-related knowledge and the convenient analysis technique [2]. Although the base knowledge is different for every field, the relevant knowledge and analysis techniques are more common. Some of them are universal or at least widely used in their field. Usually, the base knowledge is universal within the field [6].

If we are to determine the most suitable strategy for a particular situation, we need sufficient basic and related knowledge. We also need to have a convenient analysis technique. Finally, we need to be aware of the resulting efficiency constraints which control the effectiveness of the results, so we may not use these useful results in the wrong place where they will be ineffective [4].

There is a fair amount of software to assist consultants and administrators in their jobs, encompassing everything from documentation and spreadsheets to planning and dashboard tools. Many of these tools provide a specific service used as an instrument in a particular situation [7]. Some tools promise to deliver more comprehensive utilities for the business or for a specific aspect of that business. Other tools provide a means of communication, so the founders and owners may contact a professional for consultancy services. Researchers have shed light on the need to improve models for better business intelligence architecture by suggesting a convenient unified model [7, 8].

This paper aims to provide a universal model that can be implemented in consultancy and management services in a decision support system to enable the end users to get the best advice. Being universal in this context means the ability of the model to achieve its goals irrespective of the field of implementation. In other words, we intend to fulfill a more abstract BI model that suits all types of business and BI applications.

To achieve this goal, we use the following methodology. We start by observing the state-of-the-art BI modeling in the related work. Then, based on the findings, we build a universal and unified BI model. After that, we implement the proposed BI model in a MAS structure and describe the agents' offline and online state providing the needed algorithms. Finally, we apply the findings to solve an example case study.

This paper contains six sections. Section 1 is the introduction. Section 2 is about related works of the most relevant literature. In Section 3, we define the concepts we use in our model. In Section 4, we will construct and process the model for the proposed multiagent platform. In Section 5, we will discuss a case study. Finally, we will end with a conclusion in Section 6.

2. Related Work

This section scouts some of the previous research on this same topic starting with Horkoff et al. [9] which defined a strategic business intelligence model suitable for linking a business's goals with the influencers and uses separate indicators. On the other hand, our model defines the indicator as an element of the goal instead of a separate element. The influencers are distinguished in our model by the relationship between the goals and the impact of different factors. The model used in this paper is more conceptualized to suit the multiagent cooperation dealing with it.

Lavbic and Rupnik wrote a previous paper [10], which had the same goal. Their paper drew sketches for implementing multiagent systems in the organizational decision support system. That system used five agents interacting together under specific rules and ontologies. The paper used a technical approach that is demanding and hard to maintain. Our paper proposes a multiagent platform capable of providing the same services with less demanding techniques and more functionality, based on conceptual BI modeling.

In addition, a doctoral thesis at Bond University by Patrick and Loebbert [11] dealt with BI's detention and implemented it using a multiagent system. However, their methodology was precise to the example of "pricing in grocery chains." Our paper proposes a more universal and abstract cooperative model that is not defined as one type of work. Our collaborative model will conceptualize the agents' interactions.

Loebbert and Finnie [12] proposed a distributed BI multiagent framework built on a decision unit as a primary agent that controls the interaction between the data warehouse, knowledge discovery, and decision execution elements. Our paper uses a similar distributed system with more encapsulated tasking. The communication happens between agents instead of elements. Our paper also proposes a concrete technique for knowledge acquisition and transformation using a model-based approach.

Amoako [13] studied BI as a decision-making tool while using an electric company in Ghana as a case study. The study was more general and focused on determining BI's importance and whether it can be used efficiently in decision support systems. There are a few commonalities between our paper and Amoako's study considering the BI scheme. Instead, our paper adds much more critical definitions in addition to the solid modeling and processing platform.

Meanwhile, Trieu [14] raised the problem of a non-unified BI concept and revealed what is studied and what is not. His paper is a foundational work for the model in our paper, aiming to solve the problem by proposing a unified model that allows researchers to work cooperatively on introducing the implementation algorithms using the appropriate techniques.

Lans [15], in his book, wrote in detail about abstracting BI through virtualization. Technically, he proposed a data delivery platform as a new architecture to BI engines. Our paper uses a multiagent platform as an alternative. Our paper's concepts are abstracted through the model and will operate functionally in light of various factors' weight and influence.

Venkatadri et al. [16] proposed a novel BI framework based on the multiagent technology for implementing low-cost BI systems. Their paper included eight agents. Meanwhile, our paper adds another layer of abstraction to fit into three main agents and proposes the concept of the data to form a mathematically solvable model.

The cooperation paradigm is a complex one. Our model aims to create a flexible platform that enables the clients to work cooperatively while assuring their rights and privacies. These issues have been wildly searched, and counting all of them is beyond the scope of this paper. The used paradigm in this paper follows the software outsourcing partnership (SOP) [17, 18].

For the technical part of our study, many branches of artificial intelligence (AI) are involved. These can be summarized in [19–21], which sum up AI's state-of-the-art decision-making techniques considering the new challenges for big data. They also provide a summarized history of AI and decision-making through the present state of AI and describe an agenda for upcoming innovations.

This paper's proposed platform is designed using a multiagent approach similar to the architecture defined in [22]. Khozium, in his paper, defines the software agent and reveals its characteristics by describing its construction step-by-step using the case study. There is another paper focused on object-oriented programming (OOP) developers to provide the headlines to convert the OOP scheme into a multiagent system (MAS) scheme [23].

By the end of this section, we notice the need to have a universal model that satisfies the BI demand irrespective of the business type. Besides, we need to add a cooperative aspect to BI applications, so the development process will go further in every phase instead of building knowledge bases from scratch. We also find that although many MASs have been studied to serve BI platforms, these systems do not have a unified API. Considering these motivations, we have conducted this study.

We distinguish our model from the previous related works by being

(i) Universal which can support all business major
(ii) A cooperative which supports integration between encapsulated agents with service management enabled
(iii) Mathematically defined using functions that relate the goals with the factors
(iv) Based on three agents: storage agent, client agent, and analyzer agent

By satisfying the abovementioned points, this paper contributes to the BI field both on the theoretical and implementation levels. Academically, this paper supports the following research studies by establishing a robust base for solving business analytic equations and BI problems. Practically, implementing the model proposed in this paper helps the service providers manage their services and plans and help developers create universal and cooperative applications.

3. Model Concepts

In this partition, we will talk about the concepts used in our model. The structure of these concepts varies in compliance with their role in the model. Mainly, we have goals to accomplish and factors that affect the accomplishment of these goals. Each goal includes a subject part where it is considered an indicator of the fulfillment of the goal and operation on this indicator part. The factors change the fulfillment state of the goals. Each factor contains a subject, a procedure on that subject, and some constraints where the factor will be affected correspondingly.

3.1. Goal. Goals represent the final scalar for the judgment of the project. If the project fulfills its goals, it is successful. Otherwise, the project is a failure. Usually, goals are stated upon the establishment of the project. A profitable business project has many common goals, such as raising profits and maintaining a running business. Some goals are situational goals such as winning leadership in some season or setting some trend record. Looking at a goal with a more critical eye, we see that it consists of two elements: a subject and an operation on that subject. For example, if we have a goal "to raise profits," "profits" is the subject and "raise" is the operation we want to perform on that subject. We can also see that the subject might be used as a scalar to fulfill that goal. In our example, if the profit gets higher, then our goal is fulfilled. Otherwise, it is not. That is why we will call the subject of the goal as an indicator. Goals are to be presented in the model as circles. The circle contains a linguistic phrase to define the indicator and a sign to define the operation. Underneath the indicator, two squares indicate the current measurement of that status and how it would be affected by the model's factors. The default effect means that the current measurement is not expected to change. The effect value is more than the measurement, and implementing the factors in the model is expected to raise the indicators and vice versa. It is useful to use some color scheme to indicate how the goals are influenced by factors, such as using yellow where there would be no influence at all, green where the influence assents the operation, and red where the influence does not.

3.1.1. Indicators. An indicator is the subject element of the goal. It is what the goal is about. It defines the scalar and the possible operation. An indicator can be determined by finding the collective noun between the answers to the following two questions:

(1) What do we have?
(2) What do we want?

For example, a grocery owner who wants to create a goal would say, "I have a grocery shop. I want it to be the biggest grocery shop in town." The indicator here is the grocery shop.

We cannot count on the subject to be an efficient indicator without finding the aspect of indication. In our lives, we prefer taking the shortcuts and omit the aspect. For example, the aspect of profit is measured by its amount. In our earlier example, the grocery shop is measured by the business's volume, which means the shop space in that example.

Although we usually omit these aspects in our conversations, it should be clearly stated in the model to identify the indicator.

3.1.2. Operation. The operation refers to what we are looking forward to getting from the indicator. Do we want to **raise** profits? Do we want to **decrease** production costs? Do we simply want to **maintain** the number of products in stock to keep it between two thresholds? Generally, we have three operations to perform on our goals:

(1) Increase: this operation means the goal is supposed to increase the indicator value. It will be presented in the model using two plus signs (++).
(2) Decrease: this operation means the goal is to decrease the indicator value. It is presented using double minus signs (−−).
(3) Maintain: this operation means the goal is to keep the indicator value between two thresholds. It should be under the high threshold and above the low threshold. This operation will be presented using an (@) sign.

3.2. Factors. Factors are the workers, which will change the values of the goals in some way or another. A situation is a factor for the goal if its occurrence will affect the goal measurement, irrespective of what that effect is. For example, costs and revenues could be factors for profit. Production line capacity and purchase orders could be factors for the amount of product in stock.

Factors are presented as rectangles. The subject of the factor is written in that rectangle. Under that subject, a square will reveal the weight of this factor in a given context.

3.2.1. Subject. Each factor includes a subject that names the factor. If the factor is measurable, the value of that factor will directly affect the value of the goal no matter the size of that effect, which means changing the value of this factor is a subgoal. Thus, we should keep looking for the factors which affect the value of this goal. These goals are connected in the model graph using arrows with the weight written on it, indicating which goal affects the other and how much one goal will be affected by the other's influence.

For example, Figure 1 shows seeking funds as a goal. The investors are factors for this goal. The number of investors is measurable, and it directly affects the goal of "seeking funds." By virtually increasing and decreasing the number of investors, we can understand the impact they have on funding. This implies that increasing the number of investors is a goal derived from seeking funds. However, to increase the number of investors, there should be other factors, such as a good representation strategy. This could be a conference or meeting. However, good representation is not measurable. It is a factor by its existence. From the experts' advice, experience, or even searching the Internet, we may conclude that the conference would be more effective than simple representation in a meeting. This example suggests the following:

(1) We need a weighted list of factors and goals to be saved somewhere
(2) We need an analysis technique to calculate the unweighted relationships

These two points will be handled later when we talk about the information storage agent and the analyzer agent.

3.2.2. Condition. Conditions are variables that define the factor weight's efficiency in a particular model of the possible worlds, which means that some environmental variables bound the weight. The variables' values represent constraints to that weight, and their weight in the model is affected by three parameters: the goal, the factor, and the conditions.

The example in Figure 1 is real if we are in modern, financially stable countries. However, a conference will not be promising in unstable countries. This would suggest the political conditions restrict the conference weight related to increasing investors' goals, which has to be recognized somewhere in the model.

The variables affecting the efficiency of the model should be defined. With a little more investigating, we can see that the factors control the goals. This suggests that the conditions should be defined with the factors. If a variable does not affect the value of a factor in the model, we do not need to worry about it. Otherwise, it should be defined, which means that whenever we add a factor to the model, we should check its conditions and assign the proper values to the variables to ensure that it meets the constraints. Here, we come to the most basic information unit in our model, and we will discuss it further in the section on the information storage agent. We will find the relation connects the goal and the factor in light of the constraints. This relation is the weight itself.

The weight w is

$$w = \text{weight}\,(g, f, \text{dom}\,(w), \text{Cs}), \tag{1}$$

or simply,

$$w = \text{CPS}\,(g, f, \text{Cs}), \tag{2}$$

where g is the goal, f is the factor, Cs is the constraints' list, and dom(x) is the domain function.

This equation is not easy to solve with an unlimited number of constraints due to the complexity of real-world conditions. That would suggest we need a technique that eliminates as many conditions as possible. However, we only need to consider the constraints that would change the factor's weight by a reasonable amount. This will also be handled in the section on the information storage agent.

Conditions are defined in a table attached to the model, as in Figure 1.

3.3. Suggestions. When a factor involves interaction with the real world, it is wise to seek advice on successfully performing that factor. This advice proposes a method to

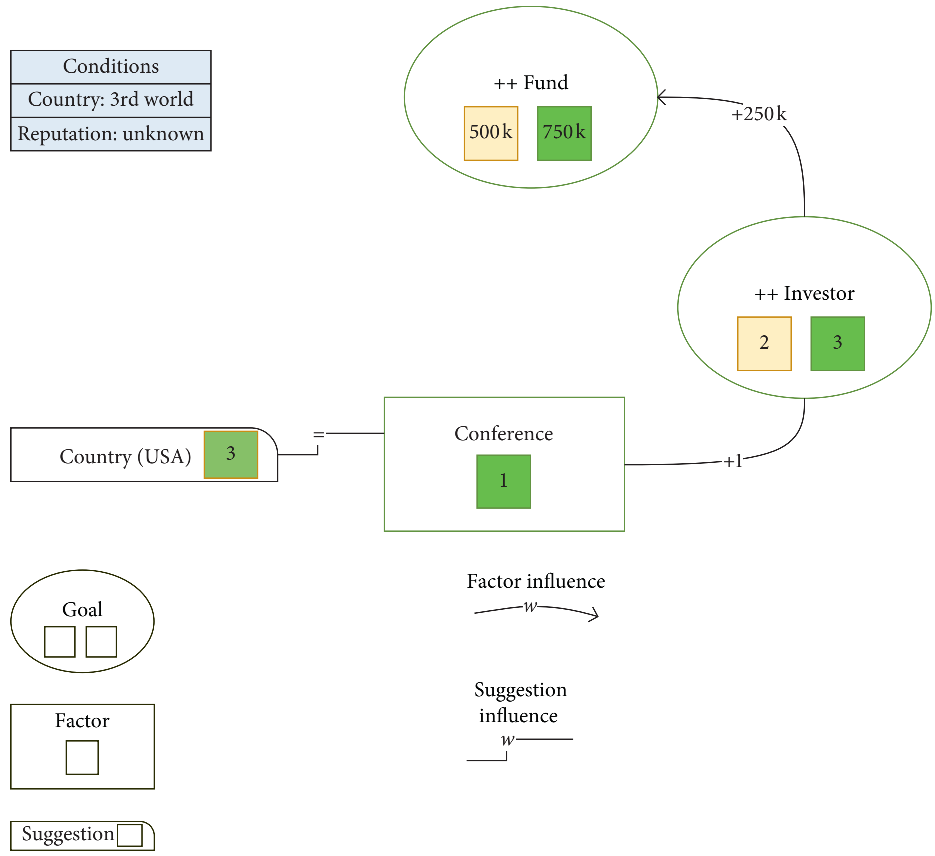

Figure 1: Example of presentation of the concepts on the model.

control a factor or many factors. For example, to hold a conference, it is useful to look at successful conferences and determine what leads to success. Proposing these factors will help determine a factor's weight efficiently.

It seems that suggestions control factors the same way that conditions do. That is correct if one adds the word "would" before the word "control." Suggestions **would** control the factors **if** they were considered. Technically, suggestions propose changing the value of a variable to obtain higher efficiency. For example, consider Figure 1 with "changing the country." The conference would yield another result. Here, it is convenient to propose holding the conference in the USA instead.

In our model, a suggestion is represented with a single round-cornered rectangle. A variable's new value is presented as a function of the variable name. The efficiency ratio is provided in a green box, which suggests that changing this variable will influence its current weight by this efficiency ratio, as shown in Figure 2. The suggestion is connected to the factor with a multiline connector to imply that it is not affected directly in the model.

If the suggestion deals with a numerical variable, it could be used as another subgoal in the model.

3.3.1. Advice. Encouraging cooperation distinguishes this model from other models. Many business situations and issues have been studied thoroughly and solved through academic research or practical experience. The results of these studies and expert decisions could be used in the same way that suggestions are used.

3.4. Influence. The influence of a subgoal of a factor on a goal is stated using an arithmetic sign and decimal number written on the line. The operations are performed on the target goal. The number represents the weight of the target's influence, while the sign will refer to what operation will be performed from the target side. For example, the division sign means the goal influence indicator will be divided by the factor value multiplied by the decimal number after the sign. The same applies to the influence of a suggestion on a factor, although most of the time, a suggestion would propose a substitutional value instead of the original factor value. In this case, an equal sign is used.

4. Proposed Multiagent Platform

For the system's infrastructure and implementation, we select the fitting AI and machine learning techniques and engines to operate this model according to the model concepts. The techniques and system should support both cooperative design and reflective architecture [24]. The cooperative design involves many other tasks, including assessing the formation [25] and dealing with its barriers [26] as constraints.

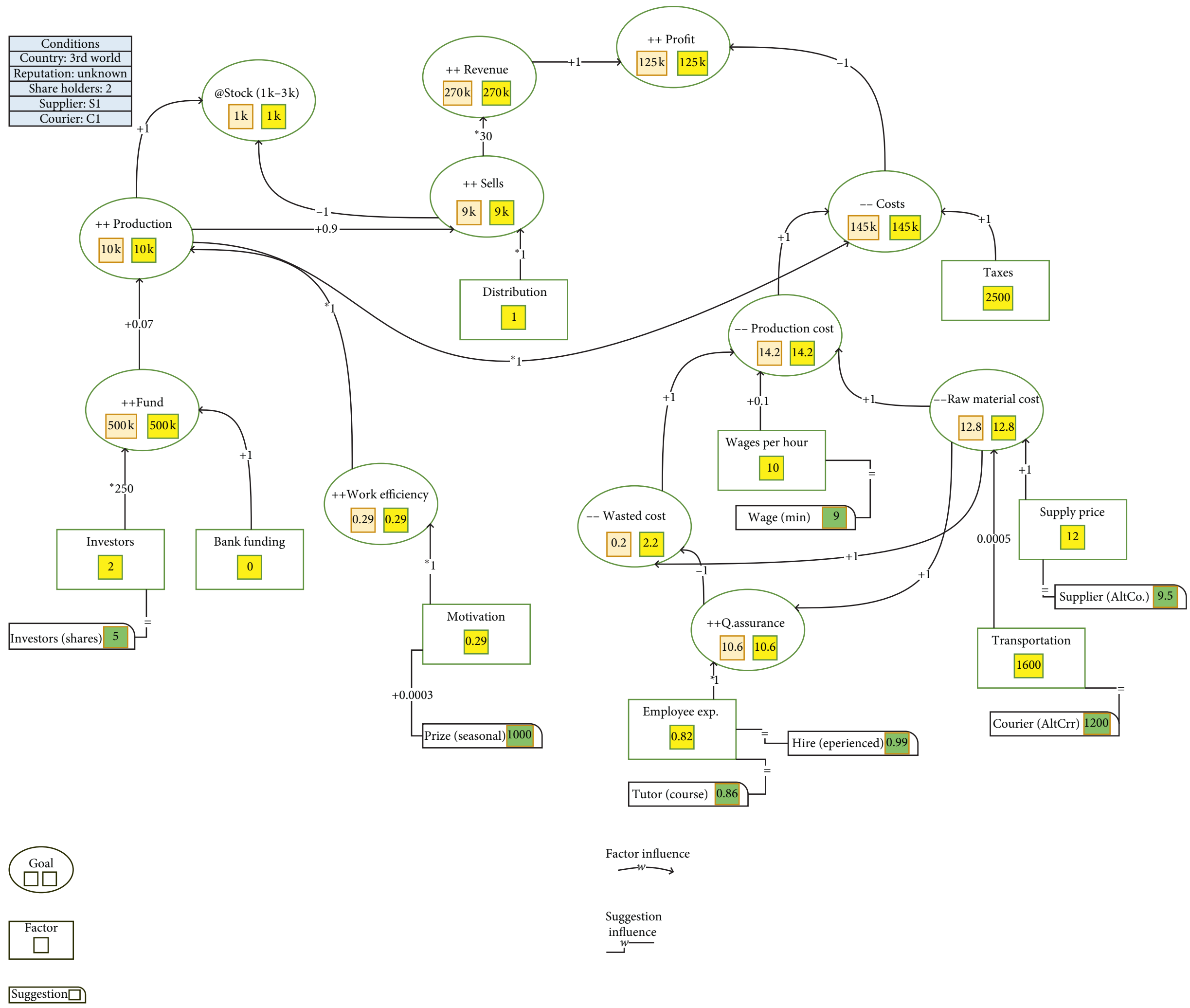

Figure 2: A state model with suggestions.

The agents analyze the concepts online, store them online, and process them online. At the same time, they update and reveal the information on the user side. This proposes using a multiagent system that engages three main agents interacting with each other asynchronously.

Figure 3 reveals the three agents and their roles. The first agent is the agent that will interact with the end user taking their inputs and reveals the results to them. The client agent, a reflective assistant agent [27], connects the BI end-user application with the universal cooperative BI model. The second agent is responsible for keeping the already processed information and making them available upon request for all other client agents. The information's availability is in the abstract form that isolates it from its primary business or exceptional cases. Meaning that the information kept by the storage agent is the rules concluded, which applies to similar situations. It is kept in the form of resolved models weighted with their relations. The weighing process and concluding the solved models are the third agent's job, which is the analyzer agent. This agent is responsible for defining the BI model's relations and weighing it to evaluate other relations later. The analyzer agent is the mind of the cooperative BI MAS that exploits the available data and operates the model's evaluation process later.

Upon implementation, the client agent is implemented by the end-user service provided using ERP, dashboard, or any decision support application. The information storage agent is a service that empowers the BI application subscription. It can be implemented using an online database with Online Analytic Processing (OLAP) enabled. The third agent is an online processing agent that has been isolated due to its overload and the need for resource distribution. It is an analytical agent that uses AI and data mining techniques to elicit the regressions and define the model's intercepts and slopes. The result will be a weighed model that is ready to support the decision-making process.

4.1. Client Agent. The client agent (CA) is responsible for collecting the business information and other required inputs from the end user (the owner or the manager) in the most suitable and convenient method for the BI program implementing this agent. The interface for this agent to the real world is the BI end-user program that implements it.

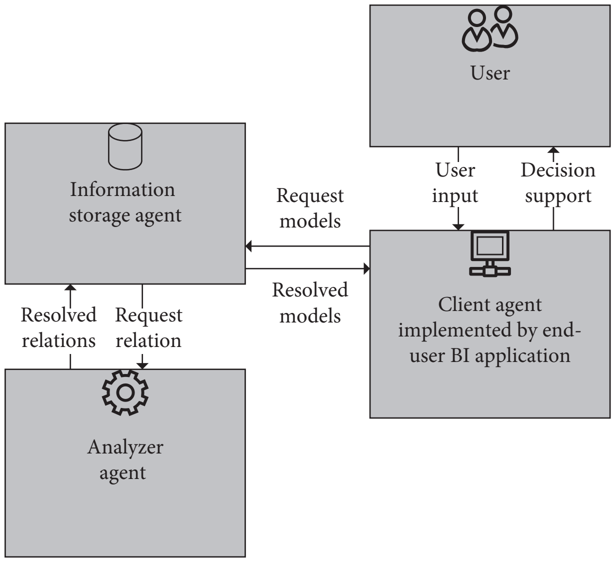

Figure 3: BI MAS interaction with the end user.

This can be a standard UI program or an intelligent program that uses other agents to interact with the end user. The CA should have a belief state regarding the business, precisely the business intelligence model we have defined above. It will obtain its belief state by requesting goals from the user. The agent will use these parameters to define the goal and create a business model in its memory. Throughout its running time, it will work on refining the model and defining the most affecting factors and proposing suggestions to control them, as graphed in Figure 4.

The developer of the CA is responsible for finding the best implementation. Using the GUI is the most popularly used approach. However, it may be more convenient to implement a speech recognition algorithm to perform an introductory conversation. The program asks the core questions to get the goals and the user's answers.

What is essential is to prepare the agent's offline information, so it is ready to run. The agent will then develop its belief state accordingly, using the available resources; this will add another offline variable that should be set before running the agent. In addition to the goals, there should be some defined knowledge resources. The resources should implement an information storage agent. At least one resource should be defined at the creation, and that will be enough to make sure that the CA will have access to all the existing information storage agents, as we will see later.

The agent will start by preparing its belief state to start a search for it in the available resources. After completing the model, the agent will try to draw the user's attention to the most effective suggestions and advice. Suppose the user agrees with a suggestion and the variable concerned is numerical. In that case, the agent will create a new goal where the variable is the indicator, and it will try to find the related factors and suggestions, as clarified in Figure 5.

4.2. Information Storage Agent. The information storage agent (ISA) is the agent responsible for providing the other agents' knowledge and updating the weights when receiving feedback. It has a belief state of an extensive list of stated goals, weighted factors, and affecting conditions. While online, it will receive goals and conditions from the CA and send back the appropriate model, which ends with the related factors that are the most efficient. To determine how efficient the weight is, we will need to distinguish between those factors, which will increase the goal value and those which will decrease it; this is done by removing one from the weight. If it is >0, then it is an increasing factor. If it is <0, then it is a decreasing factor. After that, we deal with results as probabilities, eliminating the lowest outliers from the increasing factors and the highest outliers from the decreasing factors.

It is remarkable that when the CA asks for factors and subgoals, it expects to get a list of every sufficient subgoal and factor with weights convenient with the conditions of the CA and a list of suggestions if there are any, which will negatively impact the network due to the amount of transferable data. It would be more convenient if the ISA was aware of the CA; this involves stating some conditions for the factors involved. Thus, getting information will involve at least four steps:

(1) The CA asks the storage agent for information

(2) The ISA sends a list of variables to the CA representing the condition variables so that the agent will define its value according to its environment

(3) The CA sends back the variables after assigning their values

(4) The ISA sends the model information that suits the conditions or asks for more information if needed, per adding alternative factors

This scenario will involve the end-users' input for the unassigned variables, as in Figure 6.

The previous time sequence assumes that the ISA is aware of all the factors and suggestions when considering the CA request. This assumption is not always valid. However, the ISA should be able to communicate with other ISAs to find the missing factors, which means that it should maintain a list of the other ISAs to communicate with them upon request. If an agent has access to a particular ISA, it will have access to all the ISAs.

Upon creation, the new ISA should define at least one other ISA in the ISAs' list. As it goes online, it will acquire the list from the other ISA. The agent will continuously update this list, which means the ISA's list is a current belief state for all ISAs; this would require having some central management or at least a credentialing method to protect ISAs from hacking. The second solution seems more convenient when dealing with distributed agents. Many credentialing methods can be implemented based on the variables at play; this could be such as how much the CA references the ISA or how many CAs use a particular ISA. It could use some rating methodology defined by third-party organizations. In any case, this is out of this paper's scope.

So far, we have defined the offline definition and the initial belief state. As the agent goes online, it does not have

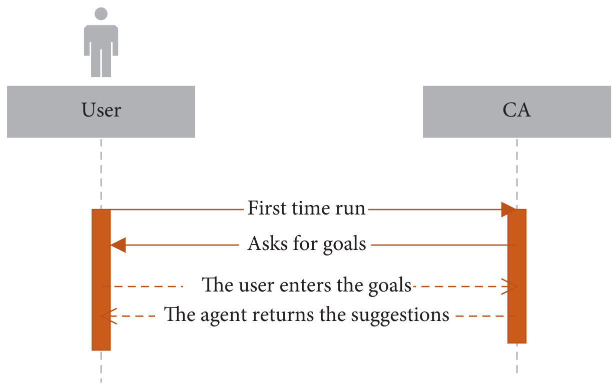

Figure 4: Interaction scenario between the end user and CA.

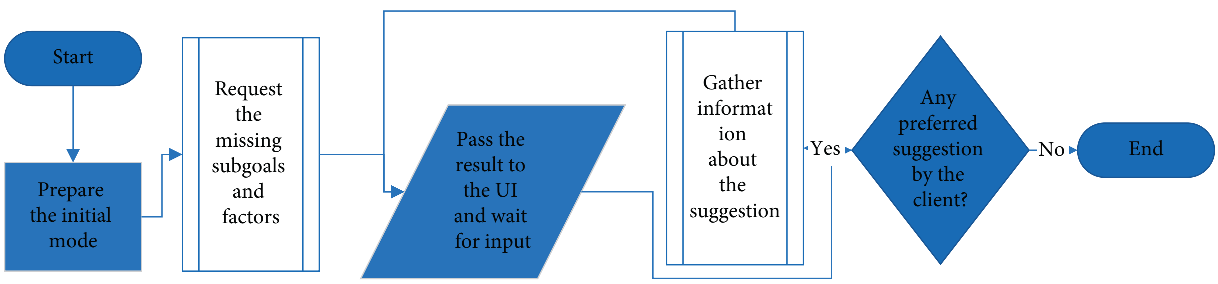

Figure 5: Client agent online flowchart.

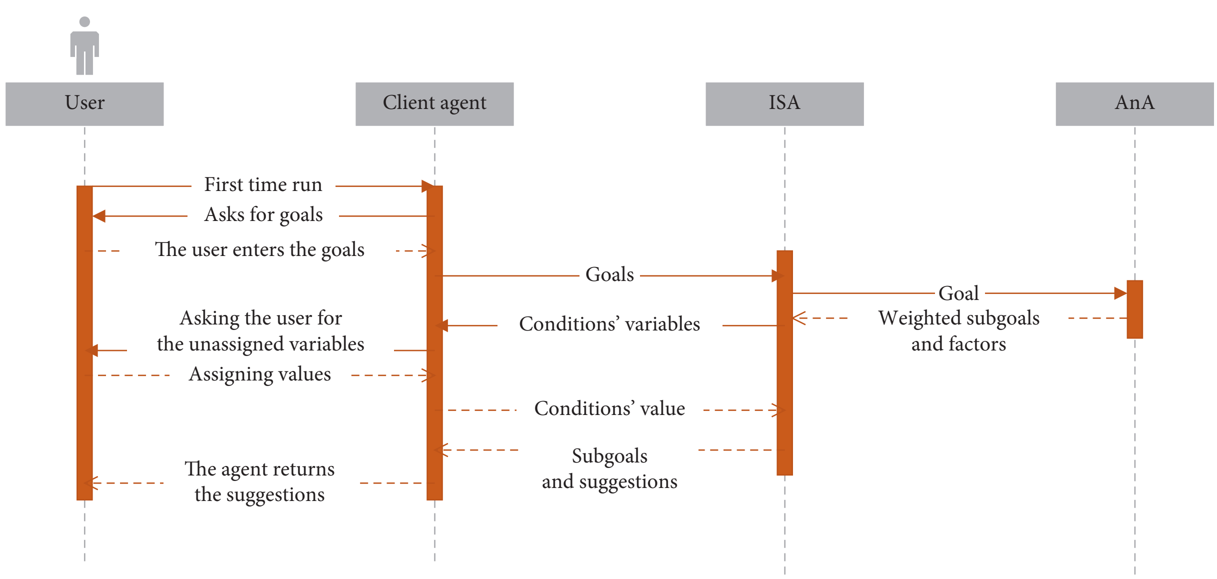

Figure 6: A general time sequence for the end user, CA, ISA, and AnA.

any knowledge. Upon request, it will try to seek knowledge for other ISAs. If the information it needs is not defined, it will request it from a third agent to analyze the goal and return the related factors and weights accordingly.

ISA online has a unique and essential mission, which is to manage the information, so it is available for the CA upon request. The belief state includes the ISAs' list and the information it obtained. While managing the relationship between the goals and factors, the quantitative factors should be considered as goals. This process will require it to determine the factors for the new goals. The goal is then sent to an analyzer agent that will return the factors and weights, as in Figure 7.

The ISA will not wait for the analyzer agent until it finishes. Instead, it will inform the analyzer about the requested information and then continue its functions. The analyzer agent will send the available results when they are ready. However, each request that the ISA has that involves

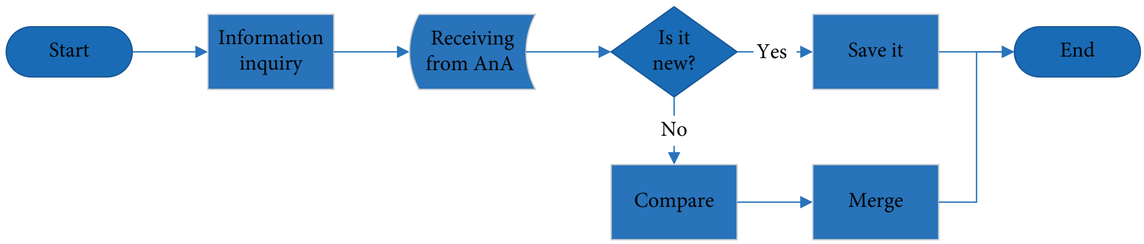

Figure 7: ISA requesting information from the analyzer agent.

the missing information will not be fulfilled until the results come in. That is why the interaction between the agents should be open and asynchronous. Each agent will act with full functionality according to its belief state. This belief state will be expanded and pruned continuously.

4.3. Analyzer Agent. The analyzer agent (AnA) is responsible for extracting the subgoals and factors of a specific goal; this can be done automatically using a hypertechnique involving a metasearch with some linguistic analytical techniques, cooccurrence detection or NLP. It can also be accomplished manually by humans based on their knowledge as academics or experts. The manual analyzer result is considered as a piece of advice. If the end user decides to use it, its subgoals will be added to the CA model.

However, automatic AnA needs more attention. It will use the indicator as a keyword to search for available content on the Internet using a metasearch. It might have been factored already by an analytical system such as Google or Facebook; this means the goal should be defined upon its creation offline.

The resulting content should be analyzed to determine the subgoals and factors. Each AnA will provide its strategy to search for the keyword and detect the subgoals and factors. The AnA will have a belief state based on the goal and what related subgoals and factors have been discovered.

The AnA will interact with the end user through the other agents. Figure 6 clarifies this interaction.

The previous time sequence tracks the results since they were requested until they were obtained. However, it is not rigorous in terms of timing. The interaction between the agents is asynchronous, as we mentioned earlier. We propose a hierarchical multiagent chart as a more meaningful chart, as in Figure 8.

5. Case Study

Figure 2 proposes a state model for a business where the main goal is to raise profits. In a realistic scenario, the profit is subject to so many direct and indirect factors. The main rule is that profit is the sum of the revenue by subtracting the total costs. In our simulated case study, we will stick to this rule. However, in a real online implementation, the ISA and AnA will recursively cooperate to form a more detailed tree based on the available connected resources, which by time, will be too comprehensive beyond the significance level. Hence, the result tree will be an expressive model that defines every variable factor that affects the goals' values. This leads us to discuss the feasibility of this model. As we simulate the proposed tree in Figure 2, applying the universal model is too much effort for such a simple calculation. Using the universal model here is kind of showing the sufficiency rather than efficiency. However, keeping in mind the online scenario, the universal model will provide a helpful tool that provides comprehensive awareness of the effective factors and their impacts on satisfying the goals. That will be the real prize, which will reveal the efficiency of this model.

According to the scenario in Figure 6, the CA will request the indicators from the implementing application and other ERP programs. It is convenient to link the indicators with individual records to update the belief state online. As the CA receives the goal "to raise profits," it will send a request to the ISA for the appropriate related subgoals and factors. The ISA might request further information to refine the model, such as the business's country, reputation, owners, product, suppliers, and couriers. Assuming that the ISA does not have the related factors for the profit indicator, it will send a request to the registered AnAs and the other ISAs. If one of the other ISAs has relevant factors, it will send them to the first ISA to construct the model. The AnA will use any available mechanism to find related factors. In our case, the AnA would use a metasearch and analyze the results' content to find that the profit equals the revenue minus the costs. The AnA then returns the goal with its factors to the ISA, where it will be formed according to the factors' conditions. These two factors are measurable and will need to be increased or decreased, so they will be placed in a subgoal format and analyzed again.

After receiving the requested information from the end user and assigning it in the model, we will have a ready-to-use state model. This state model describes the current conditions and situations of the business, along with their interactive relationships. The model is supplied with some suggestions to improve the situation in light of the goals. If the end user agrees to a suggestion, the CA will request information about the suggestion and inform the end user. There may be modifications or new goals according to the suggestion state.

In this scenario, we assume that the end user has chosen to proceed with the suggestion to raise the shareholders' number to 5. A new goal, "add more investors," will be added with new factors controlling it, such as arranging a conference and suggestions such as choosing where to arrange it. The CA will recalculate the influence accordingly, as in Figure 9.

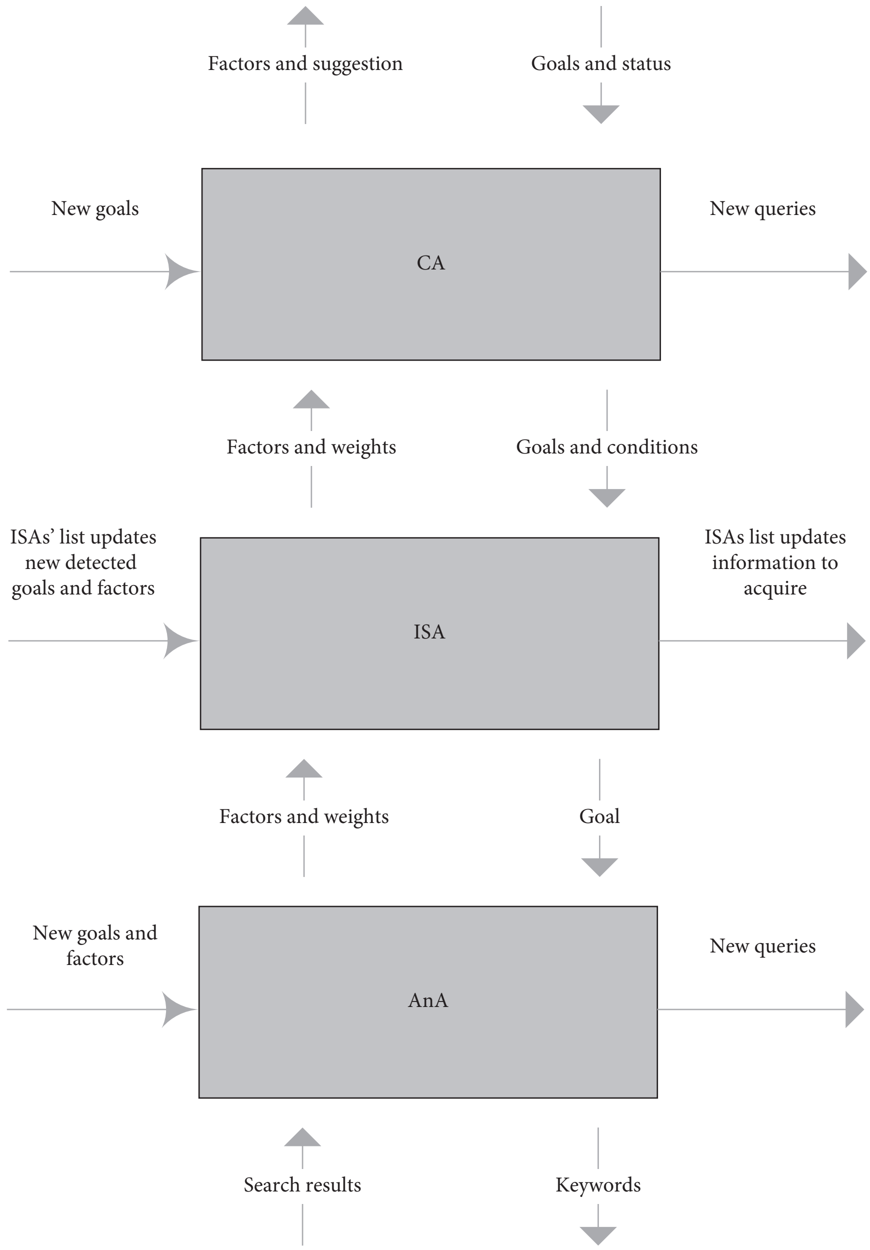

Figure 8: Hierarchical multiagent chart for the proposed platform.

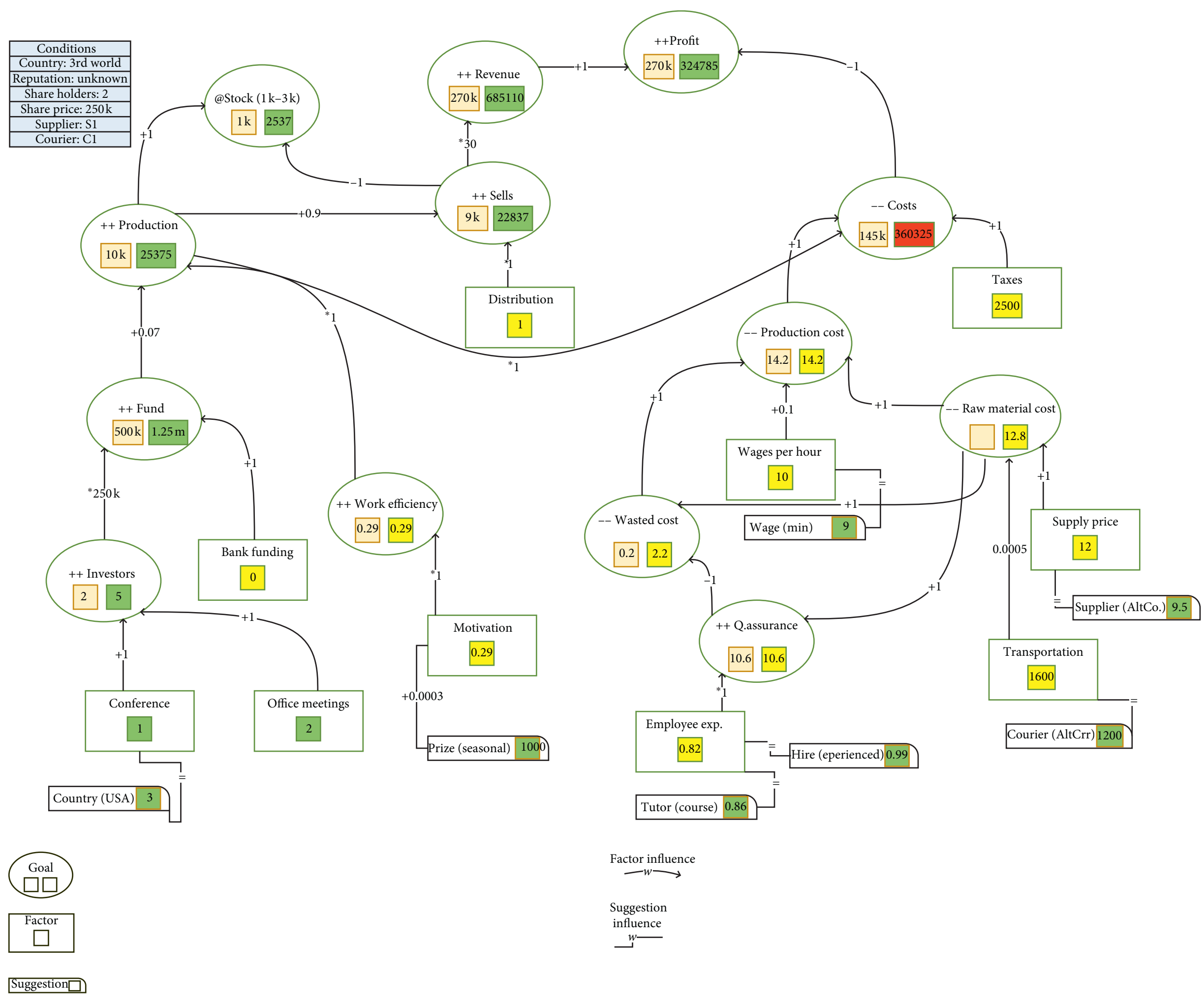

Figure 9: Recalculated model based on end-user approval.

This simulation's expected result is a tree model that shows the relationship between the goals and factors and its impact on the expected outcome in each step toward the main goal. The ISA stores this model for future retrieval. The CA will use this model to make the required calculations for the provided services, especially for proposing alternatives or providing decision support.

6. Conclusion

A cooperative business intelligence model to provide support for startups and established businesses are demanded and demanding. This paper has proposed a convenient solution that will automatically and cooperatively use the cumulative business knowledge modeled in the information storage agents, thereby providing suggestions and advice to all clients by representing that information by the client agents and resolving the unknown relations using the analyzer agent. Using the MAS to implement this model enables better overload distribution and more accurate service measurement for the service providers' financial aspect. While this paper proposes a universal and cooperative BI model, the initial implementation of this model is done quickly by configuring the available services to satisfy the basic requirements to build this model's agents. A more efficient implementation requires more effort on the cooperative work in BI in addition to constructing privacy boundaries and sharing policies that represent the future work. By now, this model is promising and opens the road to broaden the horizons in the business intelligence field by empowering universality and cooperation.

Data Availability

No formal data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] K. Grønhaug and J. Falkenberg, "Organizational success and success criteria: conceptual issues and an empirical illustration," *Scandinavian Journal of Management*, vol. 6, no. 4, pp. 267–284, 1990.

[2] A. Mohammed Abubakar, H. Elrehail, M. A. Alatailat, and A. Elci, "Knowledge management, decision-making style and organizational performance," *Journal of Innovation and Knowledge*, vol. 4, no. 2, pp. 101–114, 2019.

[3] N. Caseiro and A. Coelho, "The influence of business intelligence capacity, network learning and innovativeness on startups performance," *Journal of Innovation and Knowledge*, vol. 4, no. 3, pp. 139–145, 2019.

[4] B. Wieder and M.-L. Ossimitz, "The impact of business intelligence on the quality of decision making—a mediation model," *Procedia Computer Science*, vol. 64, pp. 1163–1171, 2015.

[5] M. Habba, M. Fredj, and S. Benabdellah Chaouni, "Alignment between business requirement, business process, and software system: a systematic literature review," *Journal of Engineering*, vol. 2019, Article ID 6918105, 19 pages, 2019.

[6] R. Srinivasan, "The management consulting industry: growth of consulting services in India: panel discussion," *IIMB Management Review*, vol. 26, no. 4, pp. 257–270, 2014.

[7] M. Elhoseny, M. Kabir Hassan, and A. K. Singh, "Special issue on cognitive big data analytics for business intelligence applications: towards performance improvement," *International Journal of Information Management*, vol. 50, pp. 413–415, 2020.

[8] P. F. Kurnia and S. Suharjito, "Business intelligence model to analyze social media information," *Procedia Computer Science*, vol. 135, pp. 5–14, 2018.

[9] J. Horkoff, D. Barone, L. Jiang et al., "Strategic business modeling: representation and reasoning," *Software and Systems Modeling*, vol. 13, no. 3, pp. 1015–1041, 2014.

[10] D. Lavbic and R. Rupnik, "Multi-agent system for decision support in enterprises," *Journal of Information and Organizational Sciences*, vol. 33, no. 2, pp. 269–284, 2009.

[11] A. Patrick and J. Loebbert, *Multi-Agent Enhanced Business Intelligence for Localized Automatic Pricing in Grocery Chains*, Bond University, Gold Coast, Australia, 2012.

[12] A. Loebbert and G. Finnie, "A multi-agent framework for distributed business intelligence systems," in *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS)*, Maui, HI, USA, 2012.

[13] B. T. Amoako, *The Importance of Business Intelligence as a Decision-Making Tool*, University of Borås, Borås, Sweden, 2013.

[14] V.-H. Trieu, "Getting value from business intelligence systems," *Decision Support Systems*, vol. 93, pp. 111–124, 2016.

[15] R. v. d. Lans, *Data Virtualization for Business Intelligence Systems*, Morgan Kaufmann, Burlington, MA, USA, 2012.

[16] M. Venkatadri, H. G. Sastry, and D. Manjunath, "A novel business intelligence system framework," *Universal Journal of Computer Science and Engineering Technology*, vol. 2, no. 1, pp. 112–116, 2010.

[17] S. Ali and S. U. Khan, "Software outsourcing partnership model: an evaluation framework for vendor organizations," *Journal of Systems and Software*, vol. 117, pp. 402–425, 2016.

[18] S. Ali, J. Huang, S. U. Khan, and H. Li, "A framework for modelling structural association amongst barriers to software outsourcing partnership formation: an interpretive structural modelling approach," *Journal of Software: Evolution and Process*, vol. 32, no. 6, 2020.

[19] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of big data—evolution, challenges and research agenda," *International Journal of Information Management*, vol. 48, pp. 63–71, 2019.

[20] J. M. Álvarez-Rodríguez, G. Alor-Hernández, and J. Mejía-Miranda, "Survey of scientific programming techniques for the management of data-intensive engineering environments," *Scientific Programming*, vol. 2018, Article ID 8467413, 21 pages, 2018.

[21] Á1 Rubio-Largo, J. C. Preciado, and L. Iribarne, "Data-driven computational intelligence for scientific programming," *Scientific Programming*, vol. 2019, Article ID 5235706, 4 pages, 2019.

[22] M. O. Khozium, "Multi-agent system overview: architectural designing using practical approach," *International Journal of Computers & Technology*, vol. 5, no. 2, pp. 85–93, 2013.

[23] A. G. Abuarafah, H. Mohammed, and M. O. Khozium, "Agent vs. object with an in-depth insight to multi-agent systems," *International Journal of Engineering Science*, vol. 4, pp. 10–17, 2013.

[24] E. Tramontana, "Managing evolution using cooperative designs and a reflective architecture," *Reflection and Software Engineering*, vol. 1826, pp. 59–78, 2000.

[25] S. Ali, H. Li, S. U. Khan, Y. Zhao, and L. Li, "Fuzzy multi attribute assessment model for software outsourcing partnership formation," *IEEE Access*, vol. 6, pp. 55431–55461, 2018.

[26] S. Ali, N. Ullah, M. F. Abrar, M. F. Majeed, M. A. Umar, and J. Huang, "Barriers to software outsourcing partnership formation: an exploratory analysis," *IEEE Access*, vol. 7, pp. 164556–164594, 2019.

[27] A. Di Stefano, G. Pappalardo, C. Santoro, and E. Tramontana, "Extending applications using reflective assistant agents," in *Proceedings 26th Annual International Computer Software and Applications*, Oxford, UK, 2002.