World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

Business-Intelligence Mining of Large Decentralized Multimedia Datasets with a Distributed Multi-Agent System

Karima Qayumi, Alex Norta

***Abstract*—**The rapid generation of high volume and a broad variety of data from the application of new technologies pose challenges for the generation of business-intelligence. Most organizations and business owners need to extract data from multiple sources and apply analytical methods for the purposes of developing their business. Therefore, the recently decentralized data management environment is relying on a distributed computing paradigm. While data are stored in highly distributed systems, the implementation of distributed data-mining techniques is a challenge. The aim of this technique is to gather knowledge from every domain and all the datasets stemming from distributed resources. As agent technologies offer significant contributions for managing the complexity of distributed systems, we consider this for next-generation data-mining processes. To demonstrate agent-based business intelligence operations, we use agent-oriented modeling techniques to develop a new artifact for mining massive datasets.

***Keywords*—**Agent-oriented modeling, business Intelligence management, distributed data mining, multi-agent system.

I. Introduction

THE most commonly used applications of new content-generation technologies for discussion forums, news sites, social networks, weather reports, wikis, tweets and transaction, are generating petabytes of data in daily usage [1], [2]. The main challenges are increasing rapidly and becoming more complex due to the handling, storage and analysis of such big quantities of data. Big data [3], [4] comprise of video, texts, sensor logs, etc., for which transactions are composed records that require an intelligent mechanisms and tools to manage very large datasets. Business analysts face the challenges of moving big data to central locations in order to merge them and apply sequential data mining algorithms. Distributed data mining (DDM) [5], [6] originates from the need of mining over decentralized data sources and applying knowledge discovery to heterogeneous data environments. Multi-agent systems (MAS) [7] can manage complex and distributed computing scenarios. Therefore, designing and developing highly distributed systems on a large scale requires a conceptual architecture using agent-based technologies. In this context, several architectures and frameworks [2], [8], [9], have been designed by using MAS in DDM algorithms. Each of them focuses on different problem domains while the respective architectures do not comprise any specific roles and responsibilities for agents.

Developing a distributed Business Intelligence (BI) generation system that comprises different types of agents with the ability to communicate, discover and access knowledge from multiple information sources, poses big challenges for researchers. There is no exact architecture and methodology for designing MAS that show each agent's function, communication and cooperation work in such system structures. Therefore, to design such complex systems including MAS for DDM processes, we use an Agent-Oriented Modelling (AOM) [10] methodology to specify the system scenarios by developing various models. Such an approach aims at the integration, interaction and interoperability for the MAS development [11].

In this paper, we fill the gap by answering the research question of how to design an architecture that emerges for managing business-intelligence generation with highly distributed, large data sets. To establish a separation of concerns, we elicit the following sub-questions. What kind of methodology is required for developing such a Business Intelligence Management (BIM) system? What is the conceptual BIM-architecture model? What is the purpose of AOM for assigning the roles and behavior of agents? What type of communication protocol is required for the interaction of agents? The resulting architecture introduces a composite system of a MAS for DDM to discover knowledge from different distributed storage locations to analyze high volume, velocity, complexity and a variety of sensitive big-data environments.

The rest of the paper is structured as follows: Section II presents the background and the challenges for designing a multi-agent based distributed data-mining architecture. Section III provides an overview of the AOM methodology and related methodologies that are used for the analysis and design phase of our proposed architecture. Section IV presents the domain analysis and detailed design of the proposed architecture for using AOM by specifying goal-, domain-, and knowledge-models. Section V describes the general architecture and the roles and behavior of various agents. Section VI focuses on agent-based communication standards and protocols that are useful in agent interactions. This section presents our proposed model that has been designed based on agent interaction and communication protocols. Section VII includes an evaluation and outlines trends for methodologies of agent-oriented

Karima Qayumi from Tallinn University, Narva Mnt 29, 10120 Tallinn, Estonia, and Computer Science faculty of Kabul University, Kabul, Afghanistan (e-mail: karima.qayumi@gmail.com).

Alex Norta from Department of Informatics, Tallinn University of Technology, Akadeemia Tee 15a, 12618 Tallinn, Estonia (e-mail: alex.norta.phd@ieee.org).

engineering and the design of processes of MAS in DDM. Finally, in Section VIII, we conclude our paper with a summary and suggestions for the future development of our research.

II. Background and Challenges

Recent research has demonstrated that the DDM area employs MAS to reduce the complexities of distributed, parallel processing in heterogonous environments [8], [12]. Section II *A* provides details about Data Mining (DM) algorithms and the agents' impact on the mining processes. Section II *B* describes several researcher efforts and challenges related to designing and developing DDM systems by using MAS techniques.

A. DDM and Agents

Data mining [13], [5] traditionally focuses on the extraction of implicit knowledge and useful information from centralized data sets. Data mining involves the use of sophisticated data analysis methods and tools to discover previously unknown, valid patterns and relationships in large data sets. For example, DDM applications include credit card fraud detection, health insurance decisions, security related real-time applications, sensor networks, distributed clustering, etc., that deal with time-critical distributed data over networks. For the data analysis of such applications, all the data are first transferred to a local storage location and then the DM algorithms are applied on the required data. Furthermore, the conveyancing of critical and large-scale data to a central location is a challenge due to many potential privacy issues. With the growth of massive distributed datasets, centralized data-mining algorithms are no longer feasible [6]. Therefore, DDM together with the Knowledge Integration (KI) [13] approach is used to identify, acquire, and utilize knowledge from external distributed sources. In heterogeneous environments, distributed and parallel processing are applied to explore useful information and patterns from massive datasets [14]. The collaborative work of DDM with parallel processing to solve many synchronization problems requires a special approach as well as tools that can intelligently manage the integration and aggregation of tasks [7]. MAS is a promising approach for solving complicated data mining tasks in parallel processing in decentralized environments [15].

With respect to DDM techniques, many researchers explore the use of agents [7], [16] to improve the integration process, adaptability, reusability and interoperability in distributed environments. For example, [14] describes several aspects of DDM that can be improved by agent technology. These features are as follows: enterprise data mining infrastructure, involving domain, and human intelligence, supporting parallel- and distributed mining, data fusion, adaptive learning, and interactive mining. The authors in [17] use a MAS architecture for both DDM and Distributed Classification (DC) systems. In [2], [7]-[9], [13], the authors focus on multi-agent based DDM systems that are a model of MAS based mining used for the improvement of KI in distributed heterogeneous and homogenous data-mining.

B. Challenges Related to Agent-Based DDM

Latest research involves many contributions for developing agent-based models by combining the DDM framework and MAS technologies [13] for the improvement of DDM performance. Here we address some papers as follows:

In [14], the authors address the infrastructural and architectural weakness of the existing DDM systems that require more flexible, intelligent and scalable support. Differently to this paper, the authors do not solve the DDM issues by developing a MAS architecture that also specifies agent activities. In [18], the author applies MAS and proposes a methodology to determine, clarify and differentiate among agents during the development of MAS by using goals and a functionalities-grouping approach. The author implements three agents: a members-manager agent, decision-assistant agent and reporter-agent on Web-applications to demonstrate the feasibility of the methodology. The author does not address a clear architecture or developing methods for agent communication and interaction. The authors in [12] focus on the design and development of a multi-agent based framework for distributed BI systems. Their aim for using agent technology in this framework is to address issues in the field of BI including the integration into business processes, reduced latencies, and decision automation. Differently from this paper, the authors do not address any agent-based methods to design the architecture and there is also no clear view of architectures that show the multi-agent functions and activities. In [8], the authors develop an agent-based parallel DM system architecture that comprises three modules: a parallel data-accessing operation, parallel hierarchical clustering, and web-based data visualization. In this architecture, the authors employ two intelligent agents that are capable of analyzing unstructured textual data. However, for novel business-intelligence applications that generate massive and varied data in highly distributed systems, it is challenging to analyze and mine useful data with merely agents.

III. Overview and Selection of A Suitable Methodology

Developing a MAS-based complex system requires a high-level agent-oriented methodology. Several methodologies exist for analyzing and designing MAS-based applications in AOM [10]. These methodologies provide a common framework of system features for specifying, designing, developing, and implementing intelligent agent systems in different domains. In the analysis phase of our proposed architecture, each agent role must be identified and their interaction models constructed. The Gaia methodology [19], [20], is known as the first complete methodology for the analysis and design of MAS. We select this methodology to model the macro (social) aspect and the micro (agent internals) aspect [21] of the MAS architecture. To capture all properties of agents in distributed systems together with their relationships to the environment in our proposed architecture, we need to define all system characteristics during the analysis and design phases. According to literature [10], based on the Gaia methodology, two other extended agent-oriented methodologies exist:

ROADMAP (Role-Oriented Analysis and Design for Multi-agent Programming) and RAP/AOR (Radical Agent-Oriented Process/Agent-Object Relationship). These two methodologies focus on the analysis of the domain and design of the models for distributed systems and support the engineering of large-scale open systems [19], [22].

The roles of agents in ROADMAP encapsulate regulations, processes, and team responsibilities in which it is possible to specify, support and constrain the behaviors of agents in systems [21]. Therefore, we use the ROADMAP methodology to assign different responsibilities to different groups of agents in our architecture. This methodology has the property to design a role model in a hierarchical style that defines each team's tasks and it describes the characteristics of individual agents in relation to the entire system. We use this property to design the role model and then the latter is input for designing the goal model of our system with related quality goals. Furthermore, the knowledge model of this methodology connects the role models with the agents' environment. By using this characteristic, the knowledge belonging to an agent's role can be revised when the expected behavior of the system or the environment changes. Additionally, we focus on designing a domain model derived from the environment and a knowledge model in the ROADMAP methodology. This methodology represents the entities of the problem domain that are relevant for the system.

The ROADMAP together with the AOR methodology allow role information to be preserved and represented at run time, since several agents can apply the same role activity simultaneously [10]. Knowledge sharing between agents that are included in the system, involves very important processes during runtime. For this reason, we use the RAP/AOR methodology that provides means for representing the partially or fully shared knowledge environment. Furthermore, RAP/AOR we apply to the automation processes of agent interactions [10], [22]. In our proposed system, the agents communicate automatically without human interaction. This methodology offers the best option to deploy automation processes between different groups of agents at different layers [10].

In summary, the discussed methodologies are more concerned with graphical descriptions of work products for the design phase. We use the descriptions as a prelude to consider the conceptual modeling of our proposed system. At the level of computational design and implementation of these methodologies, we focus on different kinds of models from various aspects. For example, a goal model to define actors in the intended system, a domain model to identify related objects in the system domain and a knowledge model to indicate the properties of objects in their respective contexts.

Based on the analysis and design phase, we next introduce the following models: the goal model, the domain model, and the knowledge models.

IV. Domain Analysis and Detail Design

We start the development of our system architecture by using the ROADMAP and RAP/AOR methodologies. Our objective in developing the models based on a multi-layered design is to explain in detail the important terms of the agents' functions and behaviors pertaining to respective system scenarios. Each role is a specific agent behavior defined in terms of permissions, responsibilities, activities and interactions.

To demonstrate the problem domain by focusing on an agent-oriented approach, we describe the functional requirement of a BIM-architecture using concepts such as goal-, domain-, and knowledge models. In our analysis and design phase, we elicit these models by considering the agents' interactions and the agents' performing effects with other agents via shared resources. To understand in detail the system context, we present each model type separately in subsections. Section IV *A* comprises the goal model that captures the functional requirements, quality goals and roles of each agent. Section IV *B* gives an overview of the domain model that represents the context in which the BI-MAS operates. Finally, Section IV *C* illustrates the knowledge model that reflects the agents' activities, parallel operations and a specification of various data elements from the DDM environment.

A. The Goal Model

In the goal model of the BI-MAS that is depicted in Fig. 1, we first present the root functional goal of *Run BI-system* with the attached role of *Stakeholder*. In AOM, the root functional goal is called the value proposition that is too complex and must therefore be further refined into manageable functional sub-goals. In the first case, the quality goals *Quick/Fast* and *Trustable* ensure that the system has acceptable performance during knowledges exploration. The main goal includes roles and sub-goals that define capacities or positions with functionalities that are needed for the BI-MAS. To achieve the goal, the system requires a specific role for each agent and also sub-goals with quality goals to represent functional and nonfunctional requirements. Here we decompose the main goal that is associated with *Present information* into smaller related sub-goals such as *Arrange schedules, Orchestrate selection processes, Dispatch to clouds, Mining data, Aggregate information,* and *Submit result.*

The *Arrange schedules* goal we decompose into four sub-goals of *Receive input data, Assign task for agents, Create assignment table,* and *Share assignment table*. Additionally, this goal is attached to the role of *Scheduler* and the quality goal of *Timely processing* that represents the responsibility of the agent for setting an assignment to other single or a group of agents based on the received input data.

The *Orchestrate selection processes* goal includes three sub-goals *Control processes* associated with the quality goal of *Up-to-date, Activate new agent,* and *Terminate nonfunctional agent*. Furthermore, this goal is attached to the role of *Facilitator* together with the quality goal of *Effective collaboration* that represents the responsibility of activation and termination of agents. The *Dispatch to clouds* goal comprises also three sub-goals *Confirm connection, Transfer agents to clouds* and *Collect agent knowledge*. We attach the role of *Dispatcher* to this goal that follows DDM roles and

other related algorithms [6] for exploring knowledge from different data sites of systems that are not covered in this model due to page limitation of this paper.

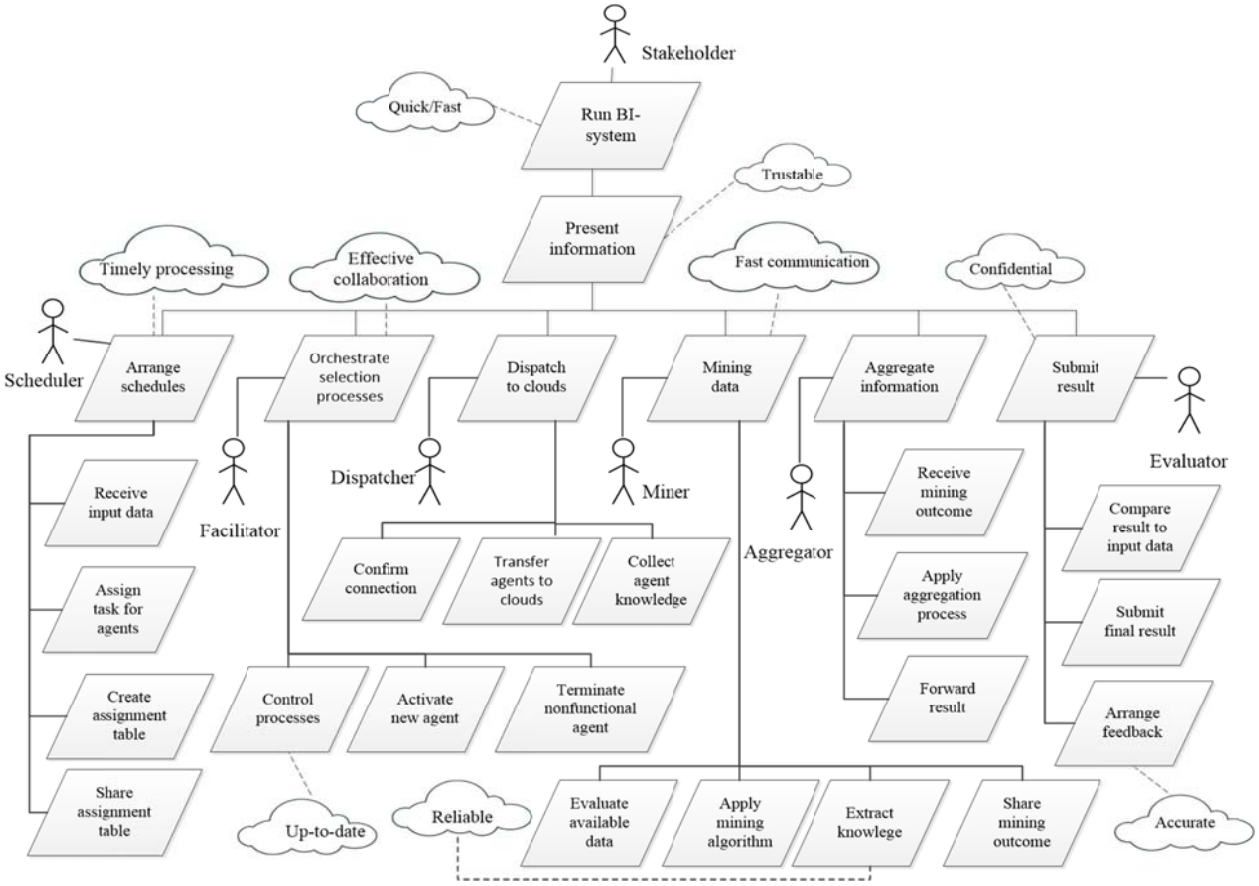

Fig. 1 The goal model of the BI-MAS

The *Mining data* goal contains four sub-goals *Evaluate available data*, *Apply mining algorithm*, *Extract knowledge* associated with the quality goal of *Reliable*, and *Share mining outcome*. The latter goal is attached to the role *Miner* together with the quality goal of *Fast communication* that represents significant performance of agents during knowledge exploration and sharing in mining processes. The *Aggregate information* goal includes three sub-goals *Receive mining outcome, Apply aggregation process*, and *Forward result.* We describe these goals with the attached role of *Aggregator* where the agent is responsible to obtain knowledge from other miner agents separately and after the collection of information forwards the result to the *Evaluator* agent.

The last *Submit result* goal comprises three sub-goals *Compare result to input data*, *Submit final result*, and *arrange feedback* associated with the quality goal of *Accurate*. This goal is attached to the role of *Evaluator* together with the quality goal of *Confidential* that represents the submitted information, which is received from a stakeholder by arranging a feedback process.

The goal model (shown in Fig. 1) and the overall process of BI-MAS can be elaborated further by applying roles and sub-goals. Due to page limitations, we cannot cover the extension in this paper. The additional portion of system analysis phase about the sharing process of explored knowledge and the agents' interaction within domain entities we discuss in the next section.

B. The Domain Model

We create a conceptual information model that represents the type of domain entities of the problem domain and the relationships among them. The model that shows knowledge about the context and the agents' interactions is called the domain model. The latter we derive from the domain entities, agent knowledge and relationships together with the context that is relevant for the BI-MAS. The domain model in Fig. 2 of our proposed architecture represents the entities of the problem domain that are relevant for a decentralized BI-MAS. The model describes the main domain entities in a multi-agent based DDM, the agents' roles, and their relationships with each other.

In this model, we consider two types of environments in which the agents are situated. The local environment is where all the activities of agents between *Data warehouse*, *System assignment table*, *Task*, *Feedback* and *Knowledge* are discussed and the distributed environment that is related to *Data sites of system*. The agent playing the role of *Stakeholder* can interact with the BI-MAS in a local environment via the

User interface. All other remaining agents are artificially made and their roles are identifiable based on their activities between these two environments. For example, the *Scheduler* agent is responsible for assigning a task for the *Miner* and *Dispatcher* agents and stores their information in the *System assignment table.* The *System assignment table* is a domain entity where all information about agents and their responsibilities are stored. Furthermore, the *Miner* agent is responsible for discovering knowledge from the *Data warehouse* that is modeled as a domain entity of the local environment. In addition, the *Aggregator* agent has a responsibility to summarize the collected knowledge from the *Miner* agent and to share with the *Evaluator* agent. Both activities of the agents belong to local environment. The *Knowledge* that is determined as new discovered information by the *Miner* agent is modeled as a domain entity in the local environment. The *Feedback* content represents the response of the *Stakeholder* agent about submitted information that is identified as a domain entity in the local environment. Furthermore, the *Data sites of system* is modeled as a domain entity in the domain model. The purpose of the *Dispatcher* agent is to explore knowledge from *Data sites of system* that is determined in a distributed environment.

The domain model of the BI-MAS (shown Fig. 2) is a compressed version due to space limitation in this paper. We cannot cover all other related domain entities and agent roles that are generated during system execution.

Next, we transform the domain model into a knowledge model to learn more about the agent's environment that is shared.

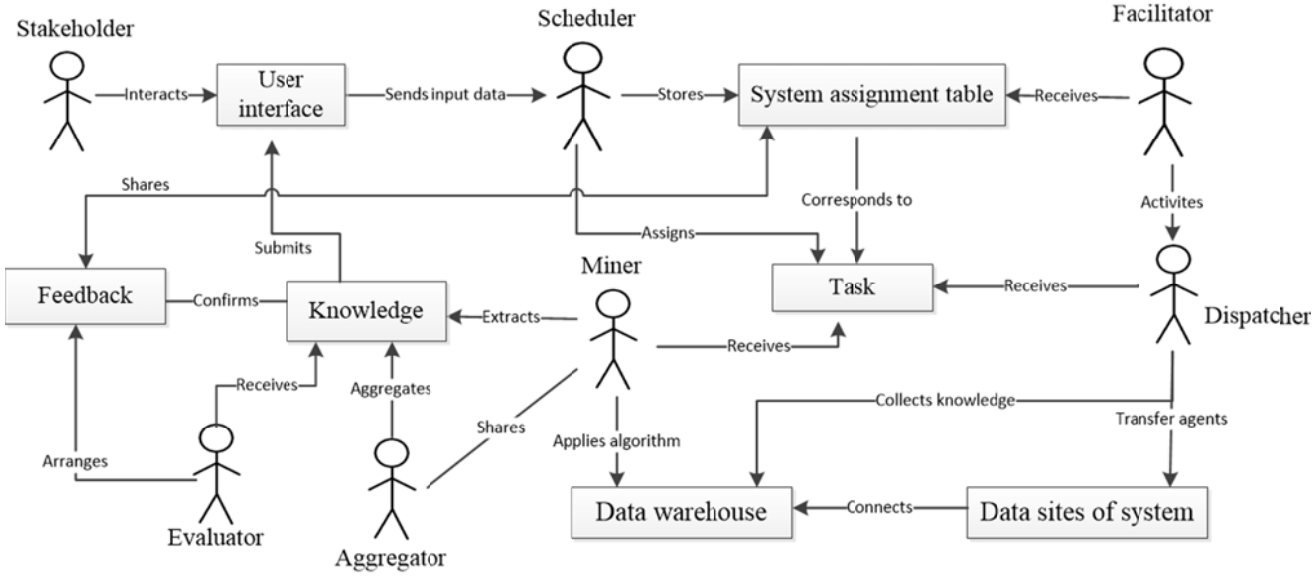

Fig. 2 The domain model of the BI-MAS

C. The Knowledge Model

We create a knowledge model of a BI-MAS that is known as an AOR agent diagram [22]. A knowledge model can be viewed as an ontology providing a framework of knowledge for the agents of the problem domain [10]. The agents' knowledge model demonstrates the agent role, internal knowledge and the relationship with objects in the environment. The new knowledge an agent represents depends on each respective role. In the knowledge model depicted in Fig. 3, each agent is modeled by representing their type, name, id, and relationships among them. For instance, the *Miner* agent knows about the task that is assigned by the *Scheduler* agent, and knows about the *DataWareHouse* where it can search to explore new knowledge. In this paper, all agent roles, relationships and their prior-knowledge are not covered in detail. Furthermore, in the knowledge model (see Fig. 3), several objects of types *SystemAssignmentTable*, *DataWareHouse*, *Data sites of system*, *Feedback*, and *User interface* are defined. These objects are shared between all agents plying different roles in the BI-MAS. For example, the *SystemAssignmentTable* in which all information about the agents' activity is stored by the *Scheduler* agent and shared between the *Facilitator*, *Miner*, *Dispatcher* and *Evaluator* agents. The *Feedback* that is followed by the *Evaluator* agent is shared between all other agents that have a specific responsibility in the local environment which is discussed before in Section IV *B*. Moreover, in each object of the knowledge model, several related attributes and predicates are defined. For example, the object of *SystemAssignmentTable* in Fig. 3 describes the attributes of *OperationId*, *InputId*, *AgentId*, *StartTime*, and *EndTime* that represent the data types and formats of data sources. Furthermore, *SystemAssignment-Table* illustrates several status predicates of agents such as *isUnscheduled*, *isScheduled*, *isInProcess*, and *isCompleted* that are followed during task assignment. Consequently, the *Feedback* object comprises of several status predicates such as *isProposed*, *isAccepted*, and *isRejected* that are used to check the *Stakeholder* satisfaction about submitted new knowledge.

In the knowledge model of Fig. 3, the number of agents, prior-knowledge, objects and relationships between agents can be elaborated for the BI-MAS as future work. In the next

section, we discuss each agent's responsibilities and roles related to the BIM-architecture.

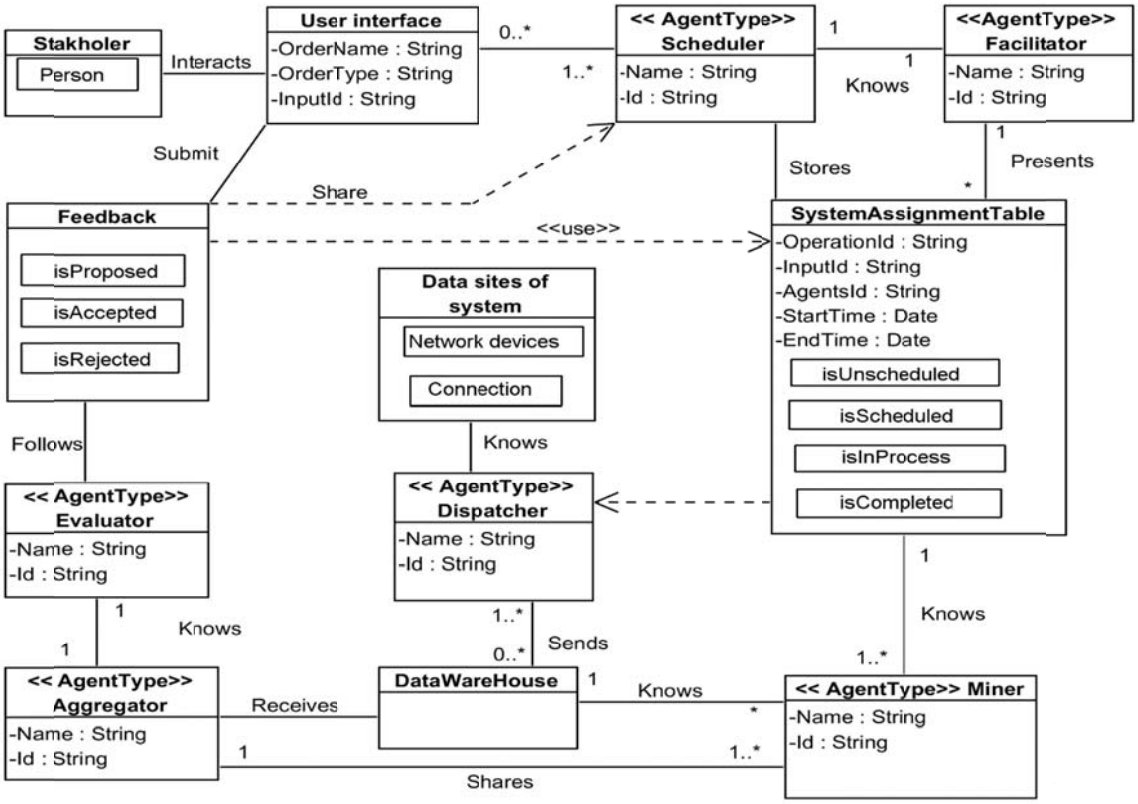

Fig. 3 The knowledge model of the BI-MAS

V. Architecture Overview and Behavior Definition

In the BI-MAS architecture shown in Fig. 4, all the earlier analysis models are transferred into a concrete layer-based architecture [23] by applying of ROADMAP together with the AOR-methodologies we discuss in Section III. Hence, roles are capabilities of the system required for achieving the functionalities [10] while the properties of a role affect the system goals. With using AOM for the BI-MAS, we define several roles such as *Stakeholder, Scheduler*, *Facilitator*, *Mining*, *Dispatcher, Aggregator*, and *Evaluator*. Each of these roles is assumed independently by an agent, or a group of agents that pursue respective goals. According to AOM [10], goals represent functionalities expected from the system and roles are capabilities of the system required for achieving the functionalities. Therefore, in this section we describe each agent role and responsibility for exploring knowledge from data in different locations. We explain all the agent roles sequentially in seven subsections, namely Sections V *A-G*.

A. Stakeholder Agent

As per Section IV *B*, the *Stakeholder* agent is defined as a human agent that interacts with the BI-MAS via the user interface (UI) to request and receive new information. In the final BI-MAS architecture of Fig. 4, this agent activity is not included. For simplicity of the complex DDM processes, we start by presenting *User interface,* that stakeholders interface with. The *User interface* module comprises functionality to capture the mining-requirement input that is shared with the BI-MAS.

B. Scheduler Agent

Scheduler agents analyze the requested input data to determine the types of mining data, operation types, and the work plan. These agents are responsible for assigning tasks to a single or group of agents based mining requirement and they also store related information such as types of mining operation, agent names and ids together with the start and end time of each agent activity to a *System Assignment Table* (see Fig. 4). This agent subsequently receives feedback based on submitted new knowledge to provide better performance on mining processes of the distributed business-intelligence.

C. Facilitator Agent

To facilitate the mining process, this agent is responsible for activating and terminating the *Miner* and *Dispatcher* agents. Moreover, this agent comprises a knowledge module that stores the history of user input data and previously retrieved information in the data warehouse that helps the *Miner* agent to apply immediate DM strategies locally [13]. In order to speed up the search process and while stakeholder submit request data, this agent automatically activates the *Dispatcher* agent to start the exploration for new knowledge from different data sites.

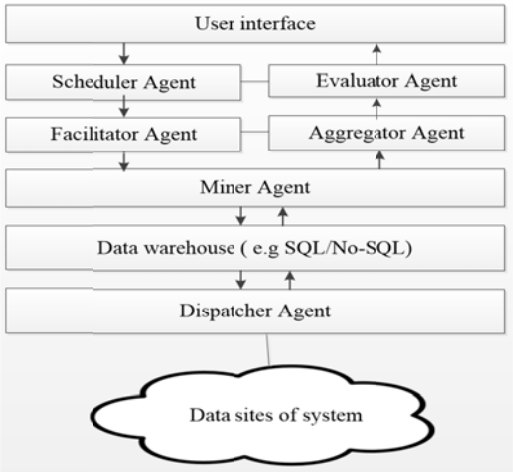

Fig. 4 General architecture of the BI-MAS

D. Miner Agent

The *Miner* agent plays an important role in the mining of data from the central physical location by deploying different mining algorithms. Additionally, this agent comprises a module to provide facilities for initiating and conducting the data mining activity, capturing the result of the mining processes and sharing it with the *Aggregator* agent. Hence, this agent has a connection with the data warehouse that plays a central role between *Aggregator* and *Dispatcher* agents.

E. Dispatcher Agent

In this layer, we use mobile agents [15] that have the capability to travel into different network locations on the Internet. The *Dispatcher* agent depicted in Fig. 4 is responsible for establishing a connection between the local and distributed environment we discuss in Section IV *B*. Furthermore, the *Dispatcher* agent is responsible for determining the computational resources at different domains and conveying the retrieved new knowledge to the data warehouse. This agent determines whether it is possible to establish parallel processes and tasks between multiple agents since the data are distributed and can be mined in a parallel manner. The parallel processing between different data sites using agents is out of scope for this paper.

F. Aggregator Agent

This agent is responsible for using BI-mining methods [14] to collect all new knowledge received from the *Miner* agent. In order to present a compact and meaningful mining result, the *Aggregator* agent plays a transformation role by resolving the conflicts and contradictions of newly mined information. Moreover, after the arrangement and integration, this agent shares the obtained knowledge with the *Evaluator* agent.

G. Evaluator Agent

The *Evaluator* agent synthesizes the final mining result together with feedback. Additionally, this agent is responsible for comparing the newly derived knowledge with requested input data to present accurate output information for BI-MAS stakeholders. Therefore, the *Evaluator* agent analyzes the responding feedback from the stakeholders about the requested BI-data and shares this with other agents that are involved in the mining processes. After the completion of the analysis and evaluation of feedback, if the *Evaluator* agent finds that the newly mined knowledge is not confirmed by stakeholders, e.g., in exceptional cases the stakeholder is not happy with newly received knowledge. Consequently, the repetition of mining processes by the BI-MAS starts with new stakeholder requirements and the latter is out of scope for this paper. In the next section, we introduce the agent-based communication protocols that are used by the BI-MAS.

VI. Agent-Based Communication Protocol and Model

Our multi-layered agent-based architecture is composed of multiple interacting agents that use synchronizations and the exchange of messages to achieve their goals and manage their tasks. In this section, we give a short description of agent-based communication protocols and their impact on BI-MAS interaction processes. Section VI *A* presents the available MAS-based communication protocols. Section VI *B* gives an example snippet for a protocol-based interaction between two agents in the proposed architecture.

A. Communication-Based Protocols

In the BI-MAS, the agents communicate with each other and they also need to follow a common set of rules and instructions during interaction processes. The specific set of communication rules is called a protocol [24]. The communication protocols are used to specify the policy that the agents follow in their interactions with each other [25]. Actually, these protocols are a set of rules that control the communication activities between several agents. Furthermore, a set of rules that regulate the interactions between agents that work together are called interaction protocols [26]. The interaction protocols define a sequence of communication messages between the agents and describe how the agents react during the interaction processes.

Recently, several research groups have developed common standards for the communication and interaction of MAS. Citation [24] demonstrates the current twenty standards for agent communication that are applied in standardizing distributed system interoperability. FIPA (Foundation for Intelligent Physical Agents) is one particular type of these standards used with agent-based distributed systems. According to [27], [25], the FIPA standard and its extended models include specifications for MAS interoperability that define interactions in terms of an Agent Communication Language (ACL). In addition, the mandatory components of the FIPA reference a model that supports different types of agent shells, multi-layered agent communication, message- and conversation management, and dynamic platform configurations. For these reasons, we only use the FIPA standard interaction protocols to describe how agents interact in the proposed approach. Furthermore, FIPA defines a set of reusable interaction patterns and processes that are generic and can be used across heterogeneous application domains and protocols [24]. According to [25], [30] Agent Petri Nets (APN) are used to design FIPA interaction protocols between agents

in distributed environments. APN defines a new formalism for modeling the interaction of entities in large MAS. In addition, APN is defined as being an oriented bipartisan graph that comprises several elements such as transitions that correspond to actions, places are the variables of the states containing tokens [26], and arcs determine the activation conditions of a transition between agents. In an APN, every transaction carries functions that manipulate the internal state and behavior of an agent (tokens) in its environments. The distribution of these tokens in agent environments is called the marking of the APN. For further explanation, Section *B* outlines an example snippet of an APN model for a BI-MAS.

B. An APN- Model Using FIPA Protocols

In BI-MAS, we focus on designing an APN model, related to the FIPA-request and inform protocols [25]. These are very simple communicative protocols that act first to establish the connection and then to pass messages from one agent to another.

In this model depicted in Fig. 5, we consider a scenario in which the *Miner* agent first confirms the connection and then informs the *Dispatcher* agent about whether the related resource based on user requested data is available in a data warehouse. In this simple communication model, we assume that the *Dispatcher* agent is in a waiting list to receive the order. *A1* pertains to the activities of the *Miner* agent and *A2* pertains to the *Dispatcher* agent.

The function *F()* in the model transfers the messages between interactive agents. The first entity name is mentioned at the top left site between brackets that indicate the transmitter and the other entity name pertains to the receiver. Thus, *A1* denotes the transmitter and *A2* the receiver. It is also possible to add more than one receiver in this function. Furthermore, in this model *m* signifies the request and *m1* signifies the accept messages of interaction flow. Additionally, here we assume that the states *P1…Pn* represent a sequence of places and the transitions *T1…Tn* denote the sequence of transitions between two agents in the model.

To establish the connection between *A1* and *A2*, *A1* sends a **request** message *(F (A1, A2))* to *A2* to perform an action *P*. *A2* receives the request message and responds with accepting the message. Furthermore, *A1* sends the message **inform** using the function *Ft (A1, A2) =<1, A1.inform, 0>*, by using *T5*. The content of the *A1.inform* changes to *1* while *A2* receives this message and completes the process. In this model context, we only present one scenario out of many that belong to our proposed BI-MAS. Fig. 5 shows an example snippet of interacting APN models that use FIPA protocols while the complete model cannot be explained here for lack of space.

To address a holistic understanding of the BI-MAS, we employ various types of agent-oriented models during analysis and design processes. In the analysis and design phase of the BI-MAS, multiple models are required to capture the functional and nonfunctional requirements of the domain. The practice of modeling, designing and implementing the BI-MAS needs to follow a specific agent-oriented methodology. In the next section, we discuss in detail an evaluation and trends of existing agent-based methodologies.

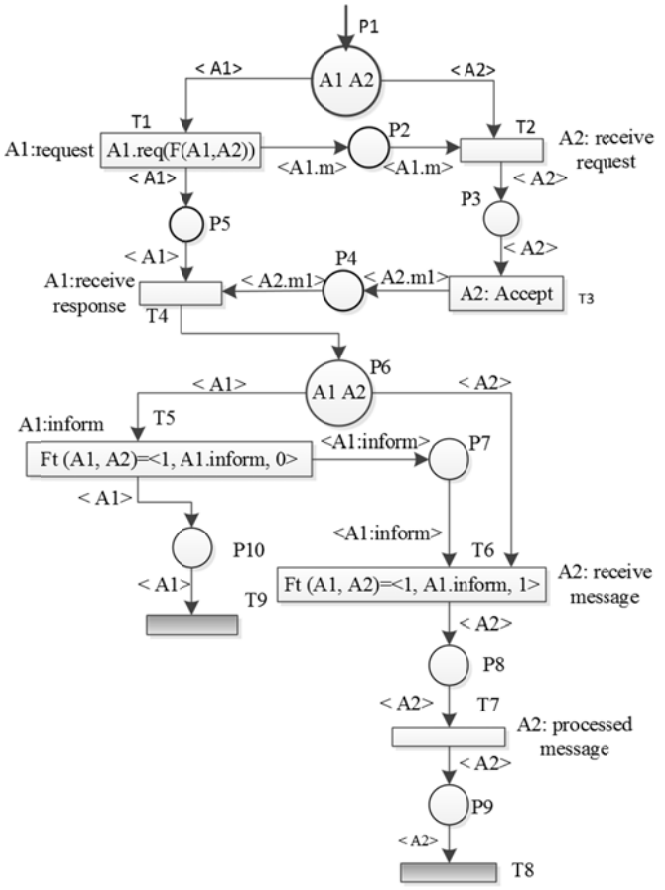

Fig. 5 A small APN-example snippet

VII. Evaluation and Trends of Agent-Oriented Engineering

For the analysis and design phase of complex BI-MAS, the role of software engineering is to provide models and techniques that assist to handle this complexity. Hence, software-agent technology provides several methodologies for building BI-MAS. Before selecting and deploying a methodology, we evaluate respective strengths and weaknesses.

Several agent-oriented methodologies and techniques such as Bayesian- and decision trees [2], Gaia and its extended approaches [19], [21] termed PASSI (Process for Agent Societies Specification and Implementation) [10] are employed to design a BI-MAS. To conduct a comparison and evaluation between these methodologies, several techniques and frame-works are proposed in literature [10], [28], which is out of scope of this paper. In order to select a suitable methodology, we consider the viewpoint framework in Table I. This framework is a counterpart to the corresponding layers of Model-Driven Architecture (MDA) [10] that proposes three types of models such as Computational Independent Models (CIMs), Platform-Independent Models (PIMs), and Platform-Specific Models (PIMs). For more detail and a better

understanding of the model, we refer reader the reader to [10]. In addition, the abstraction layers of the viewpoint framework comprise a matrix of three rows termed conceptual domain models, platform-independent models and platform-specific models.

TABLE I
THE VIEWPOINT FRAMEWORK [10]

Viewpoint models	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behavior
Conceptual domain modeling	Role models and organization models	Domain models	Goal models and motivational scenarios
Platform-independent computational design	Agent models and acquaintance models, interaction models	Knowledge models	Scenarios and behavior models
Platform-specific design and implementation	Agent interface and interaction specifications	Data models and service models	Agent behavior specifications

TABLE II
THE VIEWPOINT FRAMEWORK WITH THE MODELS OF THE ROADMAP- AND RAP/AOR-METHODOLOGY [10]

Viewpoint models	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behavior
Conceptual domain modeling	Role models (ROADMAP) and interaction-frame diagrams (RAP/AOR)	Domain model (ROADMAP)	Goal models (ROADMAP)
Platform-independent computational design	Interaction-sequence diagrams (RAP/AOR)	Agent diagram (RAP/AOR)	Scenarios and AOR behavior diagrams (RAP/AOR)
Platform-specific design and implementation	UML class and sequence diagrams (RAP/AOR)	UML class diagrams (RAP/AOR)	UML class and sequence diagrams (RAP/AOR)

We summarize that all existing agent-oriented methodologies and evaluation frameworks provide a valuable contribution to develop MAS-systems. While evaluating the Gaia process with the viewpoint framework [19], we discover that by applying this methodology, the analyst moves from an abstract to an increasingly concrete BI-MAS. This methodology is applied when the requirements of the BI-MAS are gathered and support the analysis and design phases. In Gaia, roles of agents are atomic and a construct to provide conceptual features for understanding a complex system. In this methodology, the roles are defined by specific attributes of responsibilities, permissions, activities and protocols. In addition, some specific extended features of Gaia include the ROADMAP- and RAP/AOR-methodologies that are applicable to the design of a BI-MAS. These two MAS-based methodologies enable the designer to develop an architecture with four improvements and formal models: the knowledge model with the environment, role hierarchies, an explicit representation of social structures with relationships, and an incorporation of dynamic changes [10], [21]. The combined evaluation process of Table II represents the goal-, role-, and domain models that are generated by the AOR/RAP-methodologies. Consequently, the ROADMAP focuses on application-specific domain modeling and the AOR/RAP-methodology uses certain types of UML-models during the development of the BI-MAS.

A MAS Analysis and Design Framework (MASADF) for the comparison and evaluation of agent methodologies is illustrated in [28]. The authors consider several factors during the evaluation of agent-oriented methodologies such as concepts, simplicity of visualizing a system, the models, agent attributes, the ability to represent agent interactions, agent behavior representation, and software development life-cycle points of views. In [29], the authors present an evaluation framework for agent-oriented methodologies that address six major areas: concepts, notation, processes, pragmatics, and support for software engineering. The agent-oriented techniques and methodologies are potentially powerful and represent a new paradigm for developing a BI-MAS. Pre-existing literature [28], [29] demonstrates that none of the existing agent-oriented methodologies are accepted as a standard and none of the evaluation- and comparison frameworks are suitable as a standard during the evaluation processes. With our study and evaluation, we find that the selection of a proper agent-based methodology depends on the properties of a BI-MAS and the developer's consideration.

Finally, in the next section we conclude this paper along with our research work and findings.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we focus on developing a MAS for the generation of BI that employs the mining of large datasets in distributed locations by applying a compound of agent-oriented methodologies. In this context, we use the ROADMAP- and RAP/AOR-methodologies of AOM that support the conceptual modeling, analysis and the design of a BI-MAS. Consequently, these two agent-oriented methodologies lead us to the identification of business-intelligence management goals, -roles, -protocols, and behaviors that agents adhere to. To accomplish the analysis and design phase, we represent a scheme of the BI-MAS layout by developing goal-, domain-, and knowledge models. These models yield a holistic understanding of the overall BI-MAS by including several delegated agent roles, communication- and interaction protocols in a distributed environment.

The development of the BI-MAS requires an instantiation with agent-oriented methodologies. Based on the system requirements, Gaia and its extended methodologies termed ROAD-MAP and RAP/AOR are applicable for encompassing the problem-domain realization and requirement-analysis, architecture. Furthermore, these methodologies guide us to create a conceptual BIM-architecture. The targeted conceptual architecture provides essential goal hierarchies for the BIM-architecture problem domain and the roles needed for achieving the goals.

The role of AOM is to assists us in focusing on the evaluation of the core agent-oriented methodologies. We use the AOM comparison results related to agent-oriented methodologies with different evaluation frameworks. Out of

them, we consider the viewpoint framework to select a proper methodology to develop the BI-MAS architecture. For the latter, we assign specific agents to manage newly discovered knowledge from a distributed environment. We discover that in the communication and interaction of these agents, FIPA provides the basic- and network protocols, both of which cover different types of interaction, coordination, and cooperation for the BI-MAS. Additionally, the communication protocol of FIPA covers message encoding, -encryption, and the -transportation between agents.

As future work, the BI-MAS and the conceptual models illustrated in this paper must be further elaborated and extended by considering specific properties such as flexibility, adaptability, and robustness. Thus, in future research we focus on formalizing the conceptual models regarding access levels for newly discovered knowledge and the support of exception management and compensation mechanisms when a knowledge-sharing process fails. The validation process of our proposed BI-MAS architecture we plan to accomplish with simulating and verifying processes process using Color Petri Nets and the Java Agent Development Framework. Consequently, other research directions include developing advanced APN models and -patterns using the FIPA standard for BI-MAS interaction- and communication specifications.

REFERENCES

[1] B. R. Prakash and M. Hanumanthappa, "Issues and Challenges in the Era of Big Data insight," International Journal od Emerging Trends & Technology in Computer Science(IJETTCS), vol. 3, no. 4, pp. 321-325, Aug 2014.

[2] S. Krishnaswamy, A. Zaslavsky and S.W. Loke, "An architecture to support distributed data mining services in e-commerce environments," Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, pp. 239-246, 2000.

[3] Cognizant, "Harnessing Hadoop: Understanding the Big Data Processing Options for Optimizing Analytical Workloads," no. November 2013, 2015.

[4] E. Letouzé, "Big Data for Development: Challenges & Opportunities," Global Pulse 370, pp. 1-47, 2012.

[5] A. Kumar, A. K. Tyagi and S. K. Tyagi, "Data Mining : Various Issues and Challenges for Future A Short discussion on Data Mining issues for future work," International Journal of Emerging Technology and Advanced Engineering, vol. 4, no. 1, pp. 1-8, 2014.

[6] V. S. Rao, "Multi Agent-Based Distributed Data Mining: An Overview," International Journal of Review in Computing, no. 2076-3328, pp. 82-92, 2010.

[7] S. V. S. G. Devi, "A Survey on Distributed Data Mining and Its Trends," International Journal of Research in Engineering & Technology (IMPACT: IJRET), vol. 2, no. 3, pp. 107-120, 2014.

[8] H. Kargupta, I. Hamzaoglu and B. Stafford, "Scalable, Distributed Data Mining Using An Agent Based Architecture," Third International Conference on the Knowledge Discovery and Data Mining, pp. 211-214, 1997.

[9] A. Loebbert and G. Finnie, "A Multi-Agent Framework for Distributed Business Intelligence Systems," Hawaii International Conference on System Sciences, pp. 4129-4137, 2012.

[10] L. S. Sterling and K. Taveter, The Art of Agent-Oriented Modeling, MIT Press Ebooks, 2009.

[11] M. Garoui, B. Mazigh, B. E. Ayeb and A. Koukam, "Towards to an Agent - Oriented Modeling and Evaluating Approach for Vehicular Systems Security," International Journal of Information Technology, Modeling and Computing (IJITMC), vol. 2, no. 1, pp. 37-54, 2014.

[12] A. Loebbert and G. Finnie, "A Multi-Agent Framework for Distributed Business Intelligence Systems," Hawaii International Conference on System Sciences, 45th, pp. 4129-4137, 2012.

[13] R.Hemamalini and L. Mary, "An Analysis on Multi - Agent Based Distributed Data Mining System," International Journal of Scientific and Research Publications, vol. 4, no. 6, pp. 1-6, 2014.

[14] V. S. Roe, S. Vidyavathi and G. Ramaswamy, "Distributed Data Mining and Agent Mining Interaction and Integration: a Novel Approach," IJRRAS 4, vol. 4, pp. 388-398, 2010.

[15] J. Silva, C. Giannella, R. Bhargava, H. Kargupta and M. Klusch, "Distributed data mining and agents," German Research Center for Artificial Intelligence, vol. 18, no. 7, pp. 791-807, 2005.

[16] C. Moemeng, X. Zhu and L. Cao, "Integrating Workflow into Agent-Based Distributed Data Mining Systems," Agents and Data Mining Interaction, vol. 5980, no. i, pp. 4-15, 2010.

[17] V. Gorodetsky, O. Karsaeyv and V. Samoilov, "Multi-agent technology for distributed data mining and classification," IEEE/WIC International Conference on Intelligent Agent Technology, IAT 2003, pp. 438-441, 2003.

[18] 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.

[19] M. Wooldridge, N. R. Jennings and D. Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design," JAAMAS, pp. 1-27, 2000.

[20] F. Zamboell, N. R. Jennings and M. Wooldridge, "Developing Multiagent Systems: The Gaia Methodology," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 12, no. 3, pp. 317-370, 2003.

[21] T. Juan, a. Pearce and L. Sterling, "ROADMAP: Extending the Gaia Methodology for Complex Open System," The first International joint conference on Autonomous agents and multiagent System: part 1. ACM, pp. 3-10, 2002.

[22] K. Taveter, "Towards radical agent-oriented software engineering processes based on AOR modelling," Agent-oriented methodologies, Idea Group Inc., pp. 277-316, 2005.

[23] A. Wesley, P. C. Len Bass and R. Kazman, "Software Architecture in Practice," in Second Edition, Addison Wesley, 2003, p. 560.

[24] S. Poslad, "Specifying Protocols for Multi-Agent System Interaction," ACM transaction on Autonomous and Adaptive System, vol. 2, no. 4, p. 25, 2007.

[25] B. Marzougui and K. Barkaoui, "Interaction Protocols in Multi-Agent Systems based on Agent Petri Nets Model," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 4, no. 7, pp. 166-173, 2013.

[26] G. Bel-Enguix, M. A. Grando and M. J.-. Lopez, "An Interaction Protocol for Agent Communication," CEEMAS, vol. 4696, pp. 62-72, 2007.

[27] S. Poslad, P. Buckle and R. Hadingham, "The FIPA-OS agent platform: Open Source for Open Standards," proceedings of the 5th international conference and exhibition on the practical application of intelligent agents and multi-agents, pp. 355-368, 2000.

[28] T. Abdelaziz, M. Elammari and R. Unland, "A Framework for the Evaluation of Agent-Oriented Methodologies," IEEE, 2008.

[29] Z. O. Akbari and A. Faraahi, "Evaluation Framework for Agent-Oriented Methodologies," World Academy of Science, Engineering and Technology, 2008.

[30] A. Mahfoundhi, B. Marzougui and M. Abid, "Agent Petri Nets: Theory and Application," International Journal of Sciences and Techniques of automatic control& computer engineering IJ-STA, vol. 4, no. 2, pp. 1402-1419, 2010.