Peer-to-Peer Netw. Appl. (2016) 9:328–343
DOI 10.1007/s12083-015-0339-x

Peer to peer trade in HTM5 meta model for agent oriented cloud robotic systems

Vineet Nagrath · Olivier Morel · Aamir Saeed Malik · M. Naufal B. M. Saad · Fabrice Meriaudeau

Received: 28 July 2014 / Accepted: 2 March 2015 / Published online: 31 March 2015

Abstract Cloud computing is a methodology and not a technology. Adaptation of cloud computing services for robotic applications is relatively straightforward while adaptation of underlying ideas will require a new design attitude. Cloud computing is a cost-effective and dynamic business model. Currently cloud robotics is understood as a client server methodology which enables robots utilize resources and services placed at centralized servers. These cloud servers treat robots as any other client computer offering them platform, infrastructure, process or algorithm as a service. HTM5 is an OMG MDA based multi-view meta-model for agent oriented development of cloud robotic systems. HTM5 encourages design of peer-to-peer service ecosystems based on an open registry and matchmaking mechanism. In peer-to-peer cloud robotics, a robot can trade its hardware, software and functional resources as a service to other robots in the ecosystem. The peer-to-peer trade in such systems may be driven by contracts and relationships between its member agents. This article discusses trade-view model of HTM5 methodology and its use in developing a cloud robotic ecosystem that implements peer-to-peer, contract based economy. The article also presents a case study with experiments that implement distributed artificial intelligence and peer-to-peer service oriented trade on simulated and real robot colonies.

Keywords Cloud robotics · Model driven architecture · Cloud computing · Multi-agent systems · Peer-to-peer system · Business model

V. Nagrath (✉) · O. Morel · F. Meriaudeau
Laboratoire Le2i, UMR CNRS 6306, Le Creusot, 71200, France
e-mail: vineet.nagrath@gmail.com

O. Morel
e-mail: Olivier.Morel@u-bourgogne.fr

F. Meriaudeau
e-mail: fmeriau@u-bourgogne.fr

A. S. Malik · M. N. B. M. Saad
Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, Malaysia

A. Malik
e-mail: aamir_saeed@petronas.com.my

N. Saad
e-mail: fmeriau@u-bourgogne.fr

1 Introduction

1.1 A note to practitioners

Cloud robotics is a general term to specify the use of cloud in robotic applications. Cloud is a term used to represent computer networks and thus any application where robotics utilizes a computer network to connect with other network entities is a cloud driven robotic application. It is important to remember that cloud computing is not a new computing or network technology. Cloud computing is a business model and a methodology that utilized existing technologies in a particular manner. Cloud computing is the business of offering one's resources to entities across the cloud at a price that is regulated by quality and quantity of utilized resource. When cloud robotics emerges as a new domain, it not only adapts existing cloud computing services but also adopts the cloud computing business model. A methodology for development of cloud robotic systems should have provisions to express and implement business ideas represented by cloud businesses. HTM5 is a meta-model that is designed for agent oriented development of cloud robotic systems. The HTM5 methodology has 5 views of which the

Trade-view has models for specifying designs for peer-to-peer services oriented trade in cloud robotic ecosystems. In this article we first explain the anatomical elements in HTM5 that support peer-to-peer trade in cloud robotic systems. We present a case study implemented using HTM5 methodology that studies behaviour of simulated robot colonies deploying peer-to-peer trade. A scaled down version of these experiments were repeated on physical robots. The motivation behind the current work is to showcase HTM5 as a feasible meta-model for design of peer-to-peer, service oriented trade in agent oriented cloud robotic systems.

1.2 Background

In the past decade we have seen the emergence of cloud computing as a new business model for internet based service industry. The dot-com bubble bursted in the year 2000 after which businesses moved towards virtualization and cloud computing. Cloud computing business model allows small and medium businesses to use enterprise level resources without actually buying and maintaining the hardware and human-resource. This *pay-per-use* and *scale as and when required* methodology of cloud computing made it a popular industry model. Banking, security and standardization in cloud computing increased confidence of businesses while affordable internet connected devices and mobile connectivity exponentially increased the number of users of such services. Cloud based Applications on mobile devices and integration of traditional services in the cloud brought us today to age of cloud driven mobile business ecology. The robotic community adopted the concepts and ideas proposed by cloud computing and the phenomenon of utilizing cloud services for internet enabled robots gave birth to the domain of Cloud Robotics.

1.3 Cloud robotics: an introduction

"Cloud robotics is an emerging field of robotics rooted in cloud computing, cloud storage, and other Internet technologies centered around the benefits of converged infrastructure and shared services" [22]. Making computing, service and data resources location independent enables a robot to outsource its functional requirements. Unlike in traditional robotics where all hardware and software resources have to be carried on board, in cloud robotic systems a robot can work with a minimal set of hardware and software, acquiring all additional services from other robots/devices in the cloud ecosystem. Development of a service based cloud robotic ecosystem enables manufacturers to rapidly envision and implement an idea with reduced cost. Unlike in traditional robotics, the cloud robotic ecosystem enables robots to exchange information contributing to a global pool of knowledge which is dynamically updated and is available on demand as and when required. The need to have a systematic approach towards cloud robotics is essential since the robotic/non-robotic cloud systems of near future bring a whole new canvas for cloud enabled applications. The key challenge would be to integrate the multitude of cloud enabled devices in a manner that preserves their individual design paradigm while enabling them to exchange services across any member of the cloud robotic ecosystem. Integration with the existing industrial developmental models and computation independent design tools are essential steps in making cloud robotics a feasible business model.

1.4 Cloud robotics: a business methodology

Cloud computing is a business methodology and is a remoulded use of existing computing and internet technologies. There are two levels at which cloud robotics could evolve from cloud computing. Cloud services like cloud storage, software and platform as a service (IaaS, PaaS, SaaS) could be readily adopted to work for robots by treating robots as any other clients to these cloud services. In this form, cloud robotics is cloud computing with a robot's computer as client and the underlying tools and mechanisms for the two remain the same. In other adaptations, cloud robotics could mean that robots should be made capable to provide their physical and functional capabilities as a service to other robots and computers across the cloud. In this second form, the ideas of cloud computing are adopted as such to cloud robotics, but special tools and mechanisms will be required to implement these ideas on traditional robots. Tele-operated robotics is an example for the second form, where a robot acts as a server offering its functionalities to a remote user. In both form, there is a robotic/non-robotic server that provides its functionalities as a service to other non-robotic/robotic clients. This matches with the traditional client-server mechanism in cloud computing and could be identified as client-server like cloud robotics.

1.5 Cloud robotics as a peer-to-peer system

Cloud computing is the current generation of internet computing which is currently evolving to become a peer-to-peer system where every cloud entity is a potential service provider. The next generation peer-to-peer cloud computing will give rise to a business ecosystem where entities in a cloud could freely share their resources as services without a centralized cloud server. Peers provide resources to other peers and reduce cost and dependency on original service provider. Addition of new peers increase the demand of existing resources but they also contribute to the pool of resources shared between all the peers. Most robots are entities that are capable of performing a physical action. In essence every robot is a potential service provider and a peer-to-peer cloud robotic framework is a more suitable methodology for robots working in a common physical environment. The shift from the current client-server like cloud robotics to peer-to-peer cloud robotics will enable robotic ecosystems to avail cloud resources as well as contribute to the pool of cloud based services. Service oriented peer-to-peer cloud robotic methodologies could emerge to a whole new sub-domain in multi-robot systems and Distributed Artificial Intelligence (DAI) [21] systems. Design and development of these systems will require special design tools, meta-models [19] and development methodologies.

1.6 Agent oriented cloud robotics

"Software Agents are computational entities with specific roles and personal objectives working in a visible environment with other entities which may have dissimilar roles and objectives" [8]. Distributed Artificial Intelligence (DAI) [20, 21] and Multi-Agent systems (MAS) [2, 10, 11, 23] are closely related domains. The challenge in DAI is distribution of a complicated problem between multiple entities. MAS on the other hand deal with the behavioural and transactional complexities that arise in implementation of DAI ideas. Problem formulation and distribution in multi-robot systems resembles DAI applications and thus an agent oriented approach towards design and development of multi-robot systems has some distinct advantages. Agent oriented development of cloud robotic systems enables easy transfer on DAI solutions in a cloud robotic ecosystem. A typical cloud robotic ecosystem may have several robotic as well as non-robotic entities. Figure 1 shows a typical agent oriented cloud robotic ecosystem where several robotic and non-robotic entities collaborate to establish digital business ecology. Representing robotic/non-robotic entities in a cloud robotic ecosystem by representative agents enables developers of those cloud entities to have independent product development life cycles. Unlike objects, agents are autonomous closed systems and they do not release their interior structure to the outside world. Their functionality is controlled by an internal operating logic and the communication between agents is through messages. This is different than objects since they communicate through function calls. Agents are by design better suited to implement dynamically evolving business logic and thus representing business interests of cloud robotic entities through representative agents is an attractive proposition. Agent oriented cloud robotic systems also promote inclusion of Dynamic Electronic Institutions [3, 6, 9, 17] and Digital Business Ecosystem (See Fig. 1) [12, 13] in a cloud robotic ecosystem further enhancing the usability of the approach.

1.7 HTM5 meta model for agent oriented development of cloud robotic systems

The 5-view Hyperactive Transaction Meta-Model (HTM5) [14, 16] is a domain specific Meta-model for agent oriented development of cloud robotic systems (See Fig. 2). HTM5 is based on Object Management Group's Model Driven Architecture (OMG-MDA) [18] and suggests three layers of abstractions in development of meta-models (see Fig. 3). Computation Independent layer of HTM5 has a set of graphical meta-models named Agent Relation Charts (ARCs). HTM5 allows a certain degree of flexibility in the concept of agency and allows certain agents to have an object like character towards other agents. This selective release of an agent's autonomy to other agents is named as *Hyperactivity mechanisms* in HTM5. HTM5 is a multi-view model [19] and its 5 views separates concerns with respect to a particular aspect of system and component design. The trade view of HTM5 is used to specify trade relationships in a system and is designed to support peer-to-peer trade in an agent oriented cloud robotic system. Trade in HTM5 is based on relationships that exist between various agents and special *Relation Agents* are deployed in the system to implement relational trade and business logic of the system. Another set of special agents named *Merges* are used as ports where other agents could link up with an existing system. *Merges* are used to make a system open [7] and implements logic to manage flow of messages in a cloud robotic system.

1.8 Efforts towards cloud robotics

In the past few years, a number of projects have taken initiatives towards cloud robotics. These are in addition to the developments in the cloud computing domain, which too contribute to the cloud robotic ecosystem. Following are descriptions of some of the key initiatives in cloud robotics:

- 1999: Internet of Things (IoT): Internet of things [26] is a system wherein objects, animals, people and services have the ability to transfer useful data over internet without the requirement of human interaction and help. It includes any natural or manmade object that can have an IP address and the ability to transfer data using internet.
- 2008: Web of Things: Web of Things [31] is a concept to incorporate day to day physical objects into the World Wide Web by providing them with API (application programming interface). This will help create a

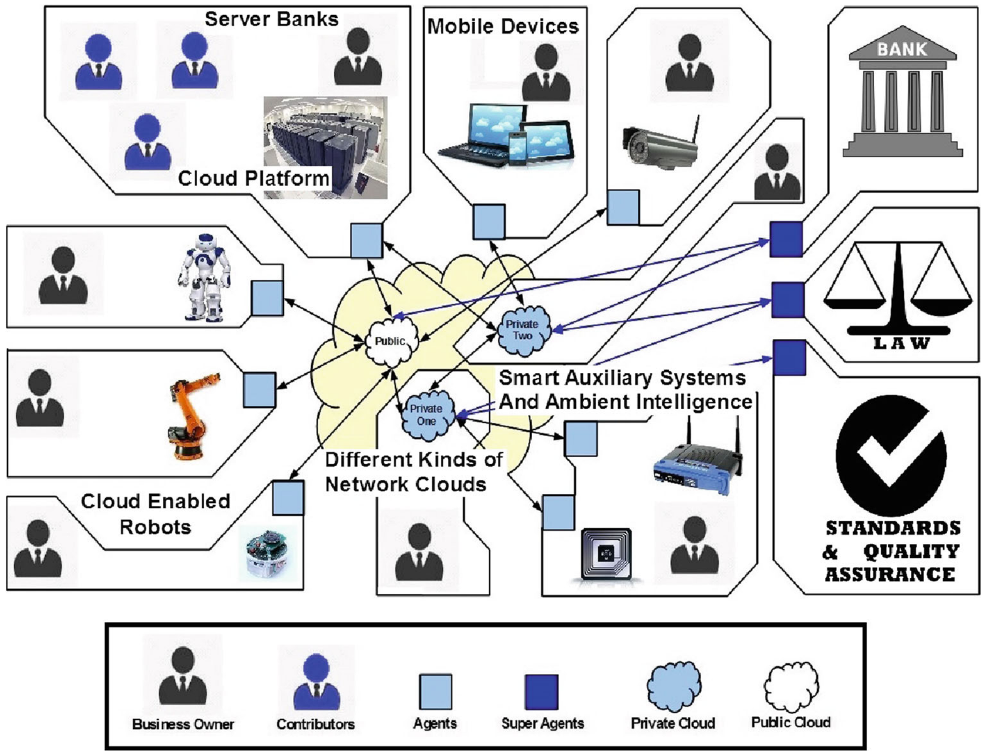

Fig. 1 Above is an example of an Agent Oriented Cloud Robotics environment that implements a Digital Business Ecosystem (DBE). In agent oriented cloud robotic system, all robotic/non-robotic entities are represented by their respective agents. The cloud entities represented by these agents may have different hardware and software configuration and may be owned by different businesses. The agents advertise the services that are offered by the entities they represent. Some cloud servers may have more than one contributor that collectively builds up a pool of resources on the cloud server. Existence of an open [7] service registry and matchmaking mechanisms enables agents advertise and enrol to services available on the cloud ecosystem. Portions of the cloud network infrastructure may be owned by private businesses and their usage may also be provided as a service in the cloud ecosystem. Special agents may be present in the system which represent the banking and administrative entities. These special agents help enforce standards (Trade, industry or legal rules) and enables transfer of money between cloud entities. Each of the entities in the cloud robotic system may have their internal developmental life cycle and the businesses that deploy these entities may have dynamically evolving business models. Entities may freely join or leave the ecosystem at will and their operating logics may change with time. An agent oriented approach enables individual entities to have their independent and dynamic operation. This ensures heterogeneity in business logic, design methodology and implementation of these entities

virtual profile for all the objects which can be used for various applications. WoT goal is to build a web of devices that is open, scalable, and flexible.

- 2010: Rosbridge [29]: This is a specification for a network layer protocol that enables communication between a ROS environment (hosted on the cloud) and a robot. ROS [28] is abbreviation for Robotic Operating System that contains tools and software libraries for robotic application development.
- 2010: DAvinCi [4]: This is a cloud computing framework for service robots. The project brings the parallelism and scalability aspects of cloud computing to the robotics domain. This project is not publically available.
- 2011: Rosjava [30]: A Java based library that allows Android devices utilize cloud services through ROS.
- 2011: RoboEarth [22]: This is a framework that allows robots to utilize database and services hosted on a WWW style server. The robots can share their behaviours with other robots through the database leading to collective learning. The purpose of this project was to prove that connection to an information network repository will catalyze the process of learning in robots that allows robotic system to perform complex tasks.
- 2011: GostaiNet [25]: This is a private project that allows sharing of vision and algorithmic behaviours amongst compatible robots.
- 2013: Cloud Based Robot Grasping [5]: A cloud robotics system for recognizing and grasping common household objects. The system utilizes Google Goggle [24] and the Point Cloud Library (PCL) [27]

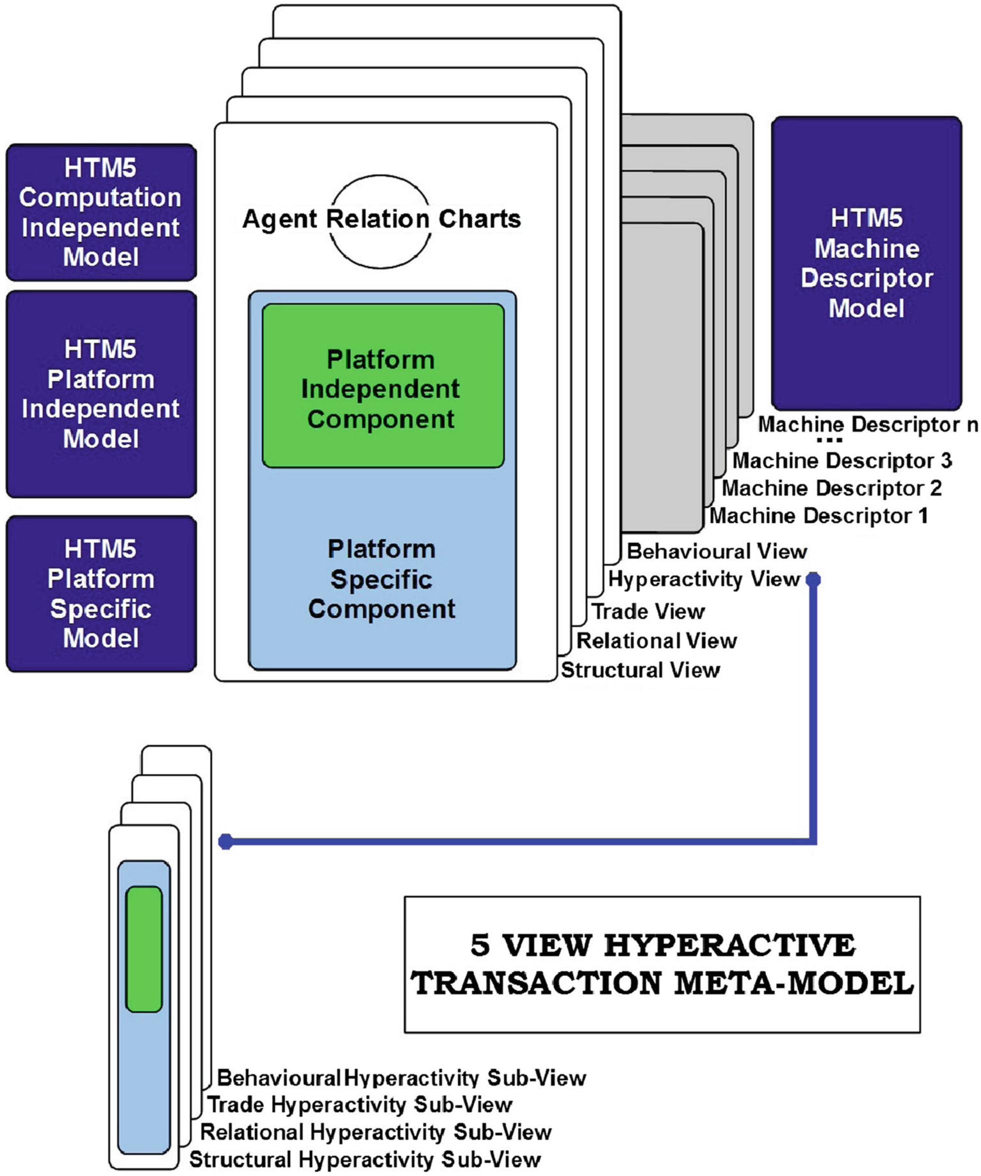

Fig. 2 An overview of 5 views Hyperactive Transaction Meta Model (HTM5) for agent oriented development of cloud robotic systems. HTM5 is based on OMG-MDA (see Fig. 3) and has a three layered structure that separate concerns of various stakeholders. The 5 views and 4 hyperactivity sub-views in HTM5 separate view-specific concerns in all three layers. Computation independent layer in HTM5 has Agent Relation Charts (ARCs) that are system level graphical models to capture structural, relational, trade, hyperactivity and behavioural aspects of the complete system. The Platform independent and platform specific layers of HTM5 are component level layers and are developed in two phases. The first phase creates class component templates for individual agent components which are then developed by various entity manufacturers in second phase of development. HTM5 also has a Machine Descriptor Model (HTM5-MDM) that models machine (host hardware or software entity) represented by an agent component

to estimate the ideal orientation for grasping common household items. This project is not publically available.

Cloud robotics as a domain is fairly new. The projects mentioned above are tools and frameworks that are an extension of traditional client server methodology. Scalability and parallelism concepts are borrowed from the cloud computing systems with robots as the clients. A subjective comparison of the popular approaches towards cloud robotic ecosystems with the Meta-Model HTM5 is presented in Fig. 4. The parameters chosen for this comparison are inspired by common needs from Industry, research and business personals.

1.9 Contribution

The HTM5 methodology [14–16] has a multitude of application domains and design features. In this paper we have given a brief introduction to HTM5 methodology. This paper does not focus on presenting the whole HTM5 methodology and all its design features. The paper is a discussion on the feasibility of the trade-view model within the HTM5 meta-model with respect to peer-to-peer contract based economy. HTM5 is designed keeping in mind the needs of researchers who wish to implement advance artificial intelligence designs on cloud robotic ecosystems. The case study experiments presented in this paper implements

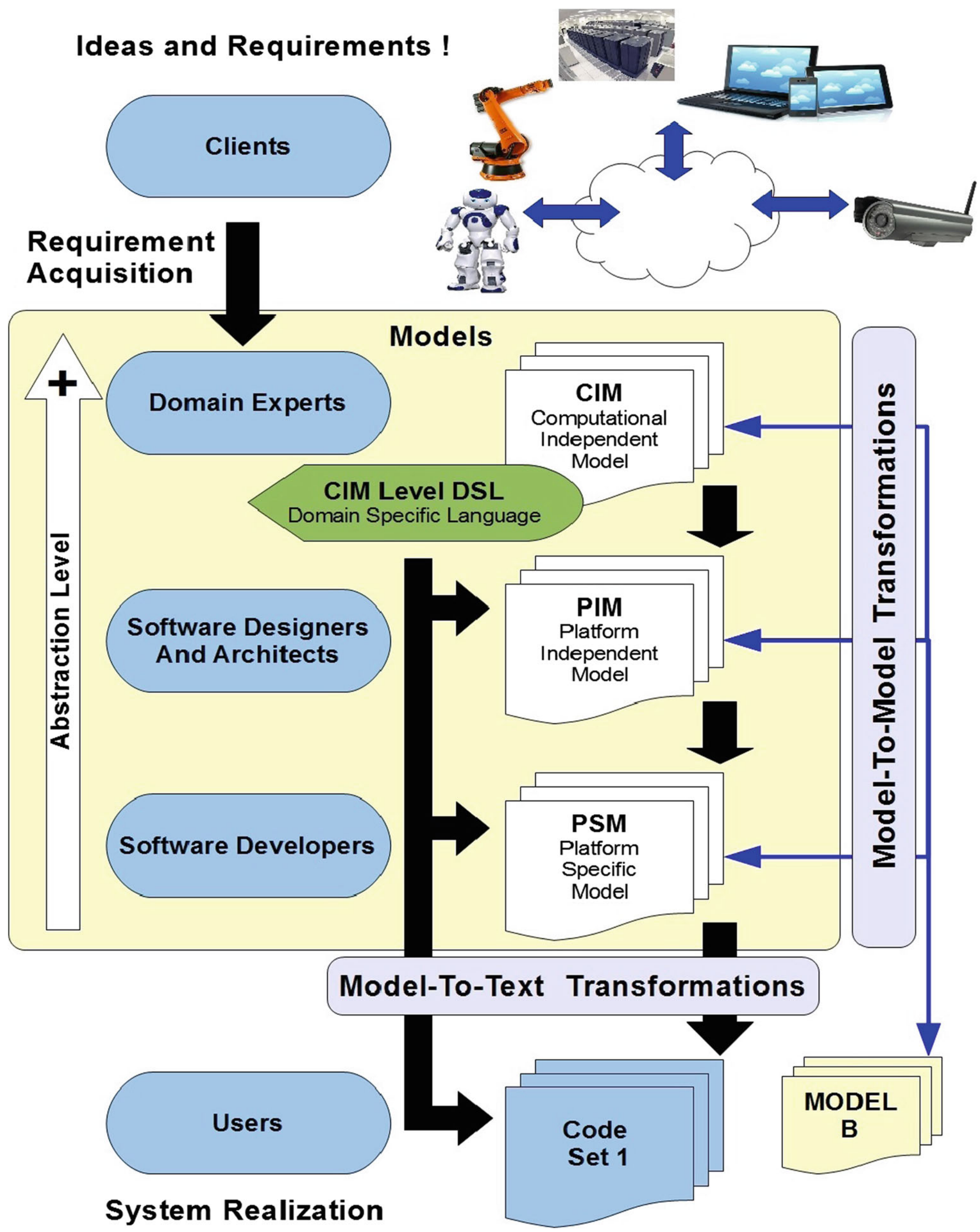

Fig. 3 Object Management Group's Model Driven Architecture (OMG-MDA) [18]. Computation Independent, Platform Independent and Platform Specific layers of OMG-MDA cater to different stakeholders in the development life cycle. Computation Independent or Platform Independent models may be supported by a Domain Specific language of same layer abstraction. The Domain Specific Language can be executed to generate automated Model to Model and Model To Text (Code) transformations

simulated and real robot colonies using the HTM5 trade modelling. The contribution of the current paper is as an implemented project for the usage of HTM5 methodology and a detailed discussion on the advantages of a peer-to-peer trade ecology in cloud robotic systems. The paper also gives the reader a step by step design walkthrough for implementing complex trade intelligence in agent oriented cloud robotic systems. The walkthrough, the case study experiments and discussion contributes to the overall usability of the HTM5 methodology.

It is noteworthy that HTM5 methodology is a methodology for development of agent oriented cloud robotic systems. These systems have robotic as well as non-robotic components. Although the focus of HTM5 is cloud robotic systems, the methodology can be used for any agent oriented cloud system with similar trade, behavioural, relational or structural anatomy.

2 Peer-to-peer trade modelling in HTM5

In Section 1 of this article we discussed the advantages of peer-to-peer cloud robotics over a traditional client-server like model. We believe that a typical cloud robotic system

	HTM5	Rosbridge	DAvinCi	Rosjava	RoboEarth	GostaiNet	Cloud Based Robot Grasping	Internet of things	Web of things	V3CMM
Is a methodology?	Yes	No	Yes	No	Yes	No	No	Yes	Yes	Yes
Is a Model driven methodology?	Yes	No	No	No	No	No	No	Yes	Yes	Yes
Follows OMG MDA guidelines?	Yes	No	No	No	No	No	No	No	No	Yes
Is specifically for cloud robotic applications?	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	
Includes auxiliary devices in the design domain?	Yes	Yes	No	Yes	No	No	No	Yes	Yes	
Is used for a holistic development of cloud robotic ecosystem design?	Yes	No	No	No	Yes	No	No	Yes	Yes	
Is not specific to a particular application?	Yes	No	Yes	No	Yes	No	No	Yes	Yes	Yes
Supports peer-to-peer operation?	Yes	Yes	No	Yes	No	No	No	Yes	Yes	
Is multi-view?	Yes	No	No	No	No	No	No	No	No	Yes
Is industry oriented?	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Has special constructs to embed business logic?	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	
Supports independent component development?	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Is Agent oriented?	Yes	No	No	No	No	No	No	No	No	
Has a machine descriptor model?	Yes	No	No	No	No	No	Yes	No	No	
is open to public use?	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Supports Dynamic Business Ecosystems?	Yes	No	No	No	No	No	No	No	No	
Supports Digital Electronic institutions?	Yes	No	No	No	No	No	No	No	No	
It is possible to use this as a Distributed Artificial Intelligence (DAI) platform?	Yes	No	No	No	No	No	No	No	No	
Has a supporting Domain specific language (DSL)?	Yes	No	No	No	No	No	No	No	No	Yes
Is the supporting DSL is computation independent?	Yes	No	No	No	No	No	No	No	No	Yes
Supports automated model to text (M2T) transformations?	Yes	No	No	No	No	No	No	No	No	Yes
Supports automated model to model (M2M) transformations?	Yes	No	No	No	No	No	No	No	No	Yes
The methodology is not dependent on associated platform and tools?	Yes	No	No	No	No	No	No	No	No	Yes
Has associated feasibility study ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Fig. 4 A subjective comparison on popular approaches towards cloud robotics and cloud connected devices

will have elements of both peer-to-peer and client-server mechanisms. A meta-model designed for modelling of such systems should enable specification of both kinds of design elements. HTM5 meta-model by itself does not inhibit or endorse a particular kind of trade model. HTM5 has 5 different views (models that separate a particular kind of design concern) of which the trade view is designed to support both peer-to-peer and a traditional client-server based trade model. In this article we present the anatomical elements of HTM5 that enables peer-to-peer trade modelling based on relationships and contracts between groups of agents. Section 3 will present a detailed case study demonstrating peer-to-peer trade ecology in a cloud robotic system.

HTM5 classifies its agents in three main categories. Agents which represent a cloud entity in the cloud ecosystem is a regular *Agent* (represented by rectangles in Agent Relation Charts) while agents which exists in the system to serve a managerial purpose are given special names and graphical representations. Agents which may/may not represent a cloud entity, but are managing relationships between other agents are called Relational agents or *Relations*. *Relations* are represented by rhombuses in Agent Relation Charts and host social and business logics of a relationship. Two or more agents may be attached to relation agents directly or through a *Merge* agent. *Merges* or merge agents are special agents which are ports to which new agents can join a cloud robotic system. The primary functionality of *Merges* is to manage the flow of messages between agents and manage the open [7] system characteristics. All three kinds of HTM5 agents (*Agents, Relations* and *Merges*) are further classified as *Active, Passive* or *Hyperactive*. This classification is based on the degree to which an agent's autonomy is released to other agents and the extent of object like character they exhibit. For the scope of current article, further discussion on *Hyperactivity* characteristics is extraneous.

Implementing cloud computing business logic in cloud robotic systems will require a mechanism for relationship based trade contracts. Out of all available services, an agent may have a business compulsion to prefer or reject certain service providers. Business logics of business owners that deploy cloud entities will require a place to exist in the system. These dynamically evolving transaction controls can be hosted in an agent's trade view class or at the *Relation* agent that is managing an agent's trade with other agents. Advance trade concepts like Digital Institutions [3, 6, 9, 17] and Digital Business Ecosystem [12, 13] could also be implemented through *Relation* agents. An open system [7] is a system where third party entities can plug in and join the system. In an open system the entities are free to join or leave the system at will. In HTM5, open systems can be implemented using *Merges*. An open cloud robotic system could define ports at which agents can dynamically join or leave the system. In HTM5 such ports could be implemented using *Merges*. *Merges* could be used to specify Ad-Hoc open systems with an unknown number of third party entities. *Merges* could also be used when within a known system, agents switch positions in relationships. The concepts and implementation of *Relations* and *Merges* is very flexible and can be used to specify any kind of trade or social dynamics. HTM5 by itself does not inhibit or endorses any particular trade methodology and provides generic tools and structure that can be used to specify any logic.

A vital functionality that can be implemented at HTM5's *Relation* agents is the service discovery and matchmaking mechanism. Peer to peer trade in cloud robotic systems will require distributed locations where service providers could advertise their services along with their associated costs and quality parameters. As relationships between trading parties are managed by *Relation* agents, it would be preferable to implement service registries and demand-supply matchmaking mechanisms on the *Relation* agents. *Relation* and *Merge* agents have more visibility in the cloud ecosystem since they are connected to a number of other agents and act as open ports. For this reason *Merges* and *Relations* are also ideal for hosting agent indexes and other trade related data items. Lookup tables for services and demands; distance, speed and quality related cost metrices and other trade variable are suitably hosted at associated *Relation* and *Merge* agents. The next Section 3 will present a detailed case study demonstrating peer-to-peer trade ecology in a cloud robotic system.

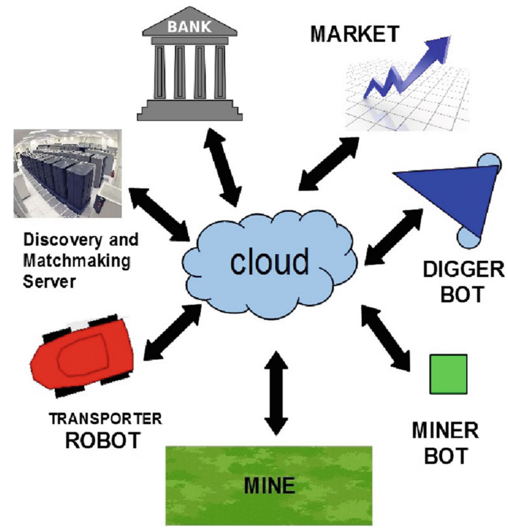

Fig. 5 The above figure shows cloud entities in the "Mine Cloud" cloud robotic system. "Mine Cloud" is a simulated agent colony representing robotic and non-robotic entities in a digital business ecosystem implemented on a cloud robotic system. The physical robots form short term collaborations to extract minerals from the mine. Miner BOT are robots which select a target mineral based on the current market prices and searches for that mineral ore in the mine field. Once the *Miner BOT* detects the target mineral, it then hires a *Digger BOT* from a pool of available digging robots based on their ground speed, service delay (digging time) and cost of service parameters. Once the mineral is dug up, another service discovery and matchmaking mechanisms associates *Miner BOT* with a suitable market (based on offered price) and a *Transporter ROBOT* (based on ground speed, loading and dumping time and cost of service). The transfer of money is managed by a banking agent which transfers service fee from one to another agent's account. All service providers advertise their quality and cost parameters in respective registries hosted on relation agents. The point of comparison in this experiment is profits made (and system's mining productivity) using a peer-to-peer trade mechanism against fixed teams of collaborating agents. The case study however primarily tests HTM5 as a feasible meta-model for implementing peer-to-peer and fixed-team based trade methodologies

3 Case study

Following are the key motivations, methods and precautions associated to these case study experiments:

1. The primary objective of this case study was to test the feasibility of HTM5 meta-model as a design methodology for implementing complex trade methodologies on agent oriented cloud robotic systems.
2. The method chosen to achieve the primary objective was to implement a cloud robotic system that implements Peer-to-Peer trade methodology with multiple trade items; service and demand advertising; service discovery; matchmaking and banking mechanisms.
3. Once the system was designed and implemented using HTM5 methodology, the secondary objective was to compare Peer-to-Peer trade methodology with some other popular methodology.
4. In this article, a Peer-to-Peer methodology for trade amongst cloud robotic entities is described as an open [7] system with multiple service providers publishing services which are matched to demands by several other cloud entities. We assumed that examining such a system against a system with fixed trade relationships (fixed teams) would be interesting.
5. Comparison of peer-to-peer trade methodology to any other methodology is a relatively subjective study as

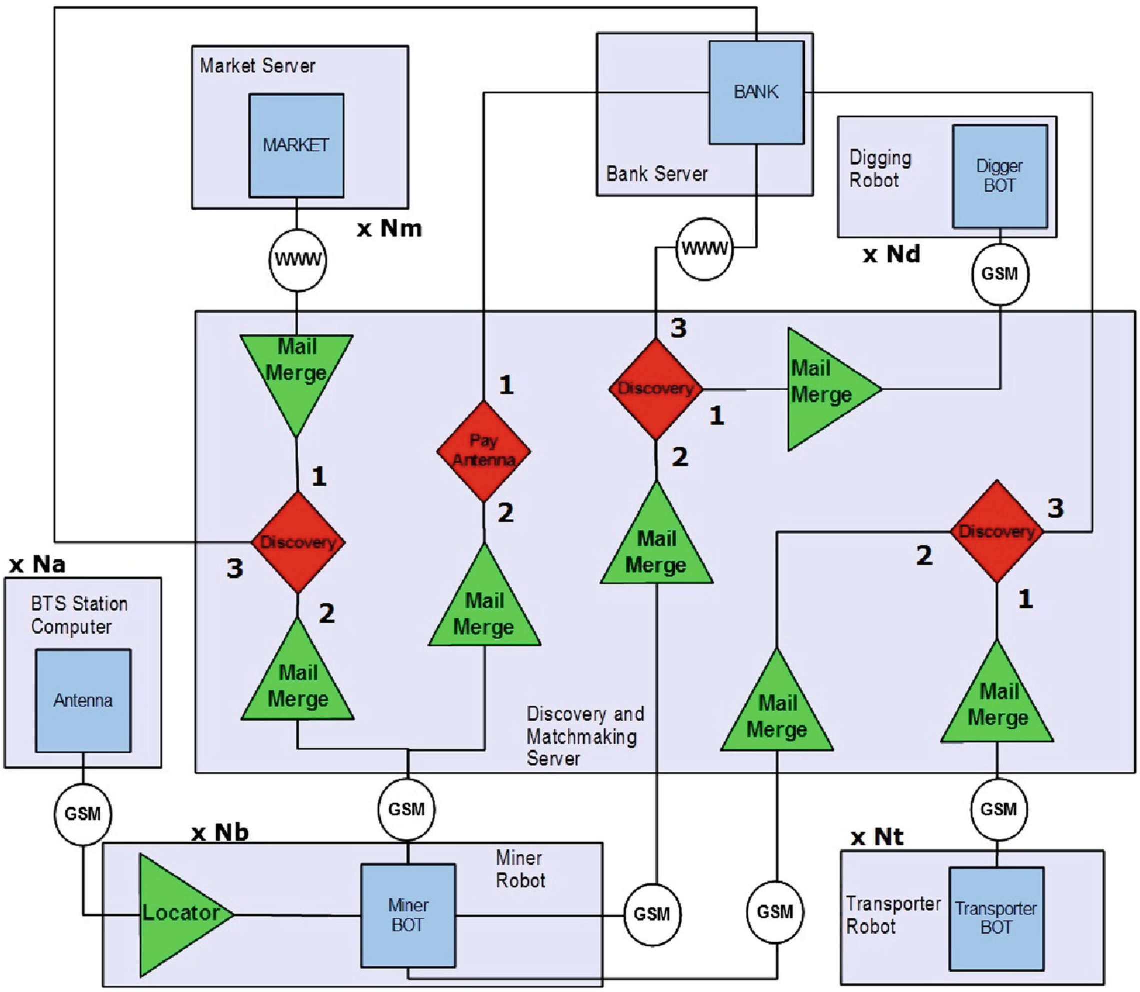

Fig. 6 Above is one of the 5 different kinds of Agent Relation Charts (ARCs) that constitutes the computation independent layer of HTM5 meta-model. Above ARC specifies the structural and relational aspects of "Mine Cloud" cloud robotic system. Above ARC specifies the physical and relative locations of different HTM5 components (*Agents, Relations* and *Merges*). Relationships between agent groups and open ports to the system are depicted by *Relation* and *Merge* agents respectively. The system simulates two different kinds of public clouds namely WWW and GSM. Cardinality of different agent groups is depicted by a number by their side. Relational cardinality (relative positioning of related agents around a relationship) is depicted by a numbered port at which they attach to a relationship

every methodology is suited for a particular scenario, ground rules and implementational realities. The observations presented in this section are based on the data collected in 32 test cases and for a particular trade environment. The authors do not claim that these results will stand valid for all trade environments in real world. The baseline idea is to implement complex trade logics on cloud robotic systems using HTM5 methodology, empowering its claim as a usable methodology for cloud robotic systems.

For the case study experiments, we envisioned a cloud robotic system named "Mine Cloud" with several robotic and non-robotic entities (see Fig. 5). Figures 6 and 7 are HTM5's computation independent meta-models for system level design specification. Computer simulations for the first part of the case study were implemented using VisuaBOT [34] and VBA [33] toolboxes. Figure 8 shows some elements of the simulating environment created for the case study. For the second part, a scaled down version of the simulated environments were implemented on a colony of five TurtleBOTs [32] (see Figs. 9 and 10). Some run time videos of the simulation experiments and experiments on the physical TurtleBOT robots are available at [1].

Following are the list of elements that constitute the "Mine Cloud" cloud robotic system: (Refer to Figs. 5, 6, 7 and 8)

Physical Entities:

1. Miner Robots **x Nb**
2. Digging Robots **x Nd**

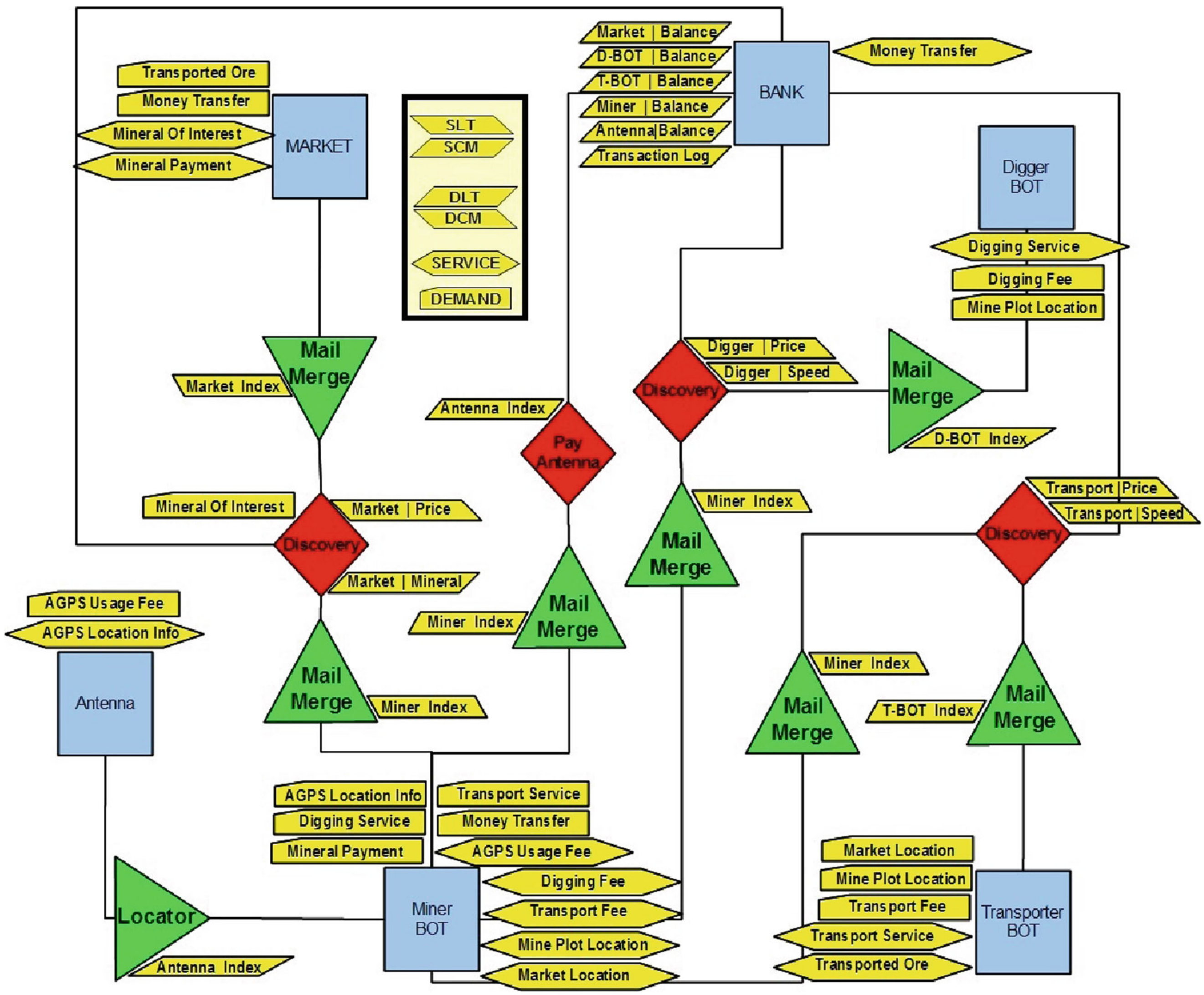

Fig. 7 Above is the Trade-Agent Relation Chart (T-ARC) for the "Mine Cloud" cloud robotic system. The structural and relational elements of this system are specified in the ARC diagram shown in Fig. 6. The Trade-ARC specifies the trade dependencies in the peer-to-peer service economy. Services, demands, lookup tables (Service LT, Demand LT), cost metrices (Service CM, Demand CM), agent indexes and trade variables are specified with respect to each trade item in this peer-to-peer trade ecology. Special *Relations* (e.g. Discovery, Pay Antenna) manage service discovery and matchmaking mechanisms and are assisted by several open ended *Merges* (Mail Merge, Locator) to allow dynamic teams

3. Transporter Robots **x Nt**
4. BTS Station Computer **x Na**
5. Bank Server
6. Market Server **x Nm**
7. Discovery and Matchmaking Server

HTM5 Components:

1. **1** *Antenna Agent* Hosted at BTS Station Computer
2. **Nb** *Miner BOT Agents* Hosted on Miner Robot
3. **Nb** *Locator Merges* Hosted on Miner Robot
4. **Nt** *Transporter BOT Agents* Hosted on Transporter Robot
5. **Nd** *Digging BOT Agents* Hosted on Digging Robot
6. **1** *Bank Agent* Hosted at Bank Server
7. **Nm** *Market Agents* Hosted at Market Server
8. **3** *Discovery Relations* Hosted at Discovery and Matchmaking Server
9. **1** *Pay Antenna Relation* Hosted at Discovery and Matchmaking Server
10. **7** *Mail-Merge Merges* Hosted at Discovery and Matchmaking Server

Cloud Networks:

1. *WWW* Between [Discovery and Matchmaking Server] And [Market Server]; [Discovery and Matchmaking Server] And [Bank Server]

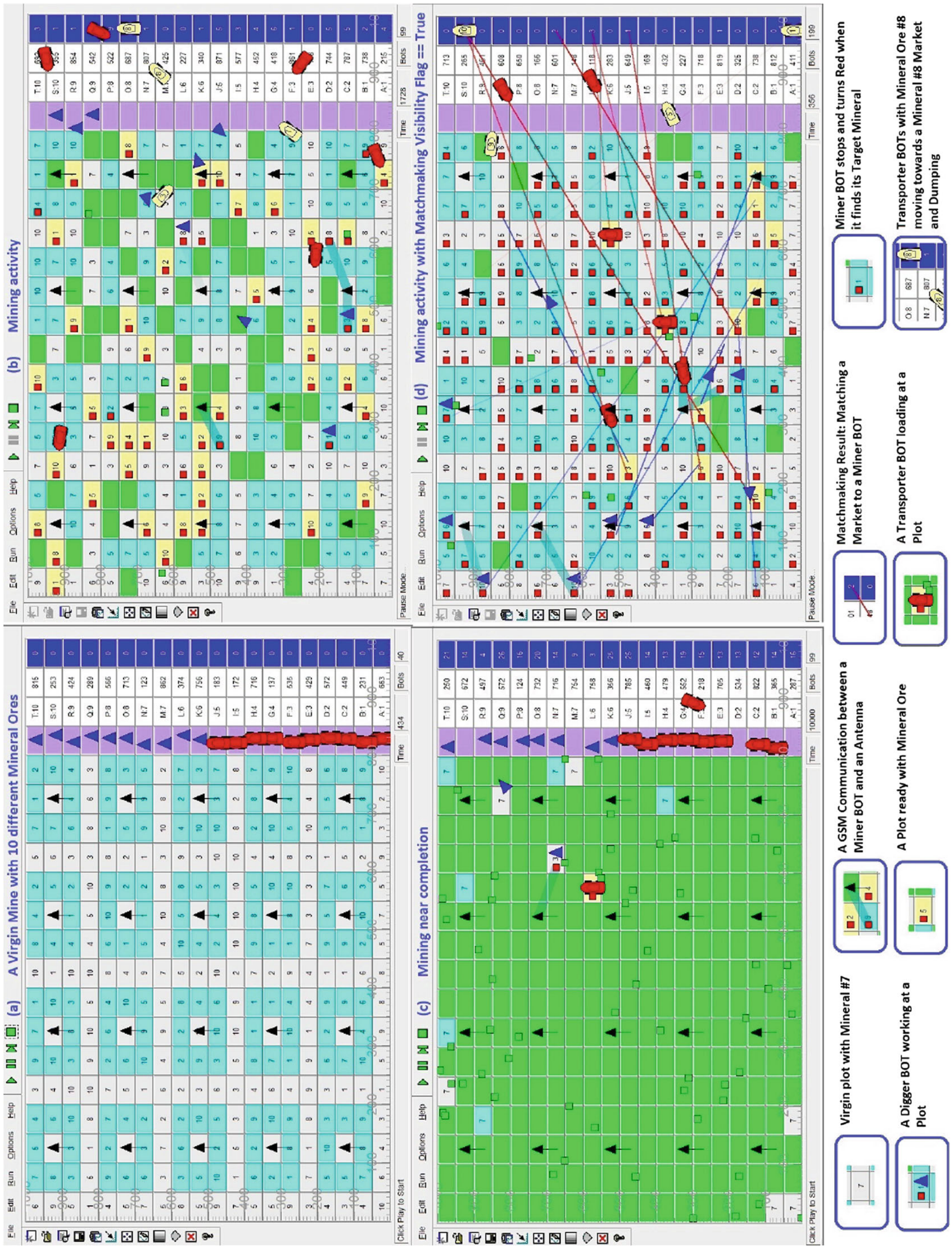

Fig. 8 Above are a set of screen shots from the simulated experiments conducted on the "Mine Cloud" cloud robotic system. The Cloud in these experiments is simulated by the inter-agent message passing mechanism of the simulator. Part (**a**) above shows the virgin mine with 10 different kinds of mineral ores randomly embedded at 320 mine plots. The right-most column of the scene simulates the physical locations of the markets and corresponding columns in white displays their preferred mineral and the cost that they offer for that mineral at a given time. At the start of the simulation, all *Digger* and *Transporter Robots* are at their parking positions. Part (**b**) of the figure shows the mining action. *Miner Robots* search for their respective target mineral ores while *Digger* and *Transporter Robots* are hired by the *Miner Robots* at different stages of the mining process. The *Transporter Robots* loads the mineral ore from a mined plot and delivers the load to the market selected by the *Miner Robot* (Through the matchmaking mechanism). Part (**c**) shows a mine field near the end of the mining process. Most of the *Miner, Digger* and *Transporter Robots* are now free as very few mine plots remains to be mined. Part (**d**) shows the mining process with matchmaking associations made visible. At the peak of mining process, the number of matchmaking associations between agents could become very frequent. Part (**e**) is a table of various parameters that could be controlled for the simulated agent colony. It is necessary to have this variability in the simulation environment so that test cases with large variability could be formed

Item	Variability											
	Count		Movement Speed		Service Delay		Service Fee		Mineral Ore Count		Mineral Ore Price	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
plot	320	320	X	X	X	X	X	X	1	10	X	X
Market	1	20	X	X	0	0	0	0	1	10	0	1000
Antenna	20	20	X	X	X	X	10	50	X	X	X	X
Digger BOT	1	10	2	20	200	500	50	150	X	X	X	X
Transporter BOT	1	10	2	20	(100, 100)	(300, 500)	50	150	X	X	X	X
Miner BOT	1	200	2	20	X	X	X	X	X	X	X	X

Fig. 9 Above is a table of various parameters that could be controlled for the simulated agent colony. It is necessary to have this variability in the simulation environment so that test cases with large variability could be formed

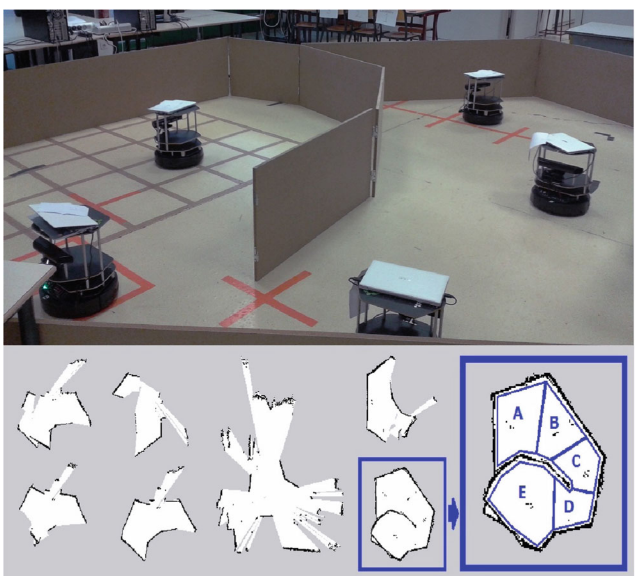

Fig. 10 The above figure shows the physical robot colony of 5 TurtleBOT robots that was used to implement a scaled down version of the "Mine Cloud" case study. Out of the five robots, three were implemented as *Miner Robots* while the other two were implemented as *Digger Robots*. No *Transporter robots* or *Physical Market locations* were implemented. There were in all five mine plots (named A, B C, D and E) which were both mining locations and parking locations for *Digger Robots*. This limited experiment was conducted to have a real life representation of the ideas tested in the simulated world. The economy (profits and productive) in physical experiments were of limited use due to very few measured parameters and limited variability. For analysis of peer-to-peer and fixed-teams trade methodologies, only data from simulated experiments was utilized. The implementation of peer-to-peer trade ideas on real life robots using HTM5 methodology was encouraging since all three layers of HTM5 design methodology was followed to write software for each of these robots in different roles

2. *GSM* Between [Discovery and Matchmaking Server] And [Miner Robot]; [Discovery and Matchmaking Server] And [Transporter Robot]; [Discovery and Matchmaking Server] And [Digging Robot]; [Miner Robot] And [BTS Station Computer];

Trade Items: Item:[Served By]–[Demand At]

1. AGPS Location Info :[Antenna]–[Miner BOT]
2. AGPS Usage Fee :[Miner BOT]–[Antenna]
3. Digging Fee :[Miner BOT]–[Digger BOT]
4. Transport Fee :[Miner BOT]–[Transporter BOT]
5. Mine Plot Location :[Miner BOT]–[Transporter BOT]
6. Market Location :[Miner BOT]–[Transporter BOT]
7. Transport Service :[Transporter BOT]–[Miner BOT]
8. Transported Ore :[Transporter BOT]–[Market]
9. Digging Service :[Digger BOT]–[Miner BOT]
10. Money Transfer :[Bank]–[Miner BOT, Market]
11. Mineral Of Interest :[Market]–[Discovery]
12. Mineral Payment :[Market]–[Miner BOT]

Trade and Matchmaking Data:

1. Antenna Index
2. Miner Index
3. Market Index
4. T-BOT Index
5. D-Bot Index
6. Market X Price (Service Lookup table)
7. Market X Mineral (Demand Cost Metrice)
8. Digger X Price (Service Lookup table)

Testcase					Dedicated Teams							Peer-2-Peer Trade							Difference					
Exp#	Rand	#Bot	#DBOT	#TBOT	#Mines	Time	Av	D Av	T Av	A Av	B Av	#Mines	Time	Av	D Av	T Av	A Av	B Av	#Mines	Av	A Avg	D Av	T Av	B Av
1	1234	10	2	2	45	5000	1638	3516	3298	1088	2029	37	5000	1659	2986	2748	1147	2199	-8	21	59	-530	-550	170
2	4321	10	2	2	28	5000	1459	2681	2056	1064	1884	27	5000	1390	2207	2135	1099	1660	-1	-69	35	-474	79	-224
3	1234	60	2	2	47	5000	1275	5315	3710	1140	1104	45	5000	1288	4925	3508	1363	1068	-2	13	223	-390	-202	-36
4	4321	60	2	2	16	5000	1112	3336	1610	1140	1011	18	5000	1123	3484	1965	1380	931	2	11	240	148	355	-80
5	1234	10	10	2	80	5000	1852	5235	1790	1136	2668	63	5000	1860	4374	1676	1187	2886	-17	8	51	-861	-114	218
6	4321	10	10	2	68	5000	1875	4512	1644	1113	3101	76	5000	1915	4294	1675	1164	3182	8	40	51	-218	31	81
7	1234	60	10	2	76	5000	1400	4986	1842	1167	1285	70	5000	1402	4470	1761	1417	1235	-6	2	250	-516	-81	-50
8	4321	60	10	2	51	5000	1276	3580	1437	1202	1197	49	5000	1287	3458	1519	1437	1127	-2	11	235	-122	82	-70
9	1234	10	2	10	47	5000	1503	1510	3329	1088	1962	38	5000	1496	1425	2786	1153	1993	-9	-7	65	-85	-543	31
10	4321	10	2	10	22	5000	1230	1273	1796	1054	1425	31	5000	1401	1312	2388	1107	1880	9	171	53	39	592	455
11	1234	60	2	10	50	5000	1276	2145	3958	1148	1085	44	5000	1257	1799	3648	1397	1040	-6	-19	249	-346	-310	-45
12	4321	60	2	10	17	5000	1100	1681	1584	1144	973	17	5000	1105	1631	1704	1356	913	0	5	212	-50	120	-60
13	1234	10	10	10	78	5000	1803	1858	1740	1141	3135	87	5000	1930	1986	1981	1229	3222	9	127	88	128	241	87
14	4321	10	10	10	71	5000	1799	1760	1662	1132	3308	109	5000	2186	2068	1986	1210	4456	38	387	78	308	324	1148
15	1234	60	10	10	206	5000	2065	3576	3279	1322	1859	206	5000	2034	3279	3309	1558	1773	0	-31	236	-297	30	-86
16	4321	60	10	10	188	5000	1936	2908	2844	1394	1803	192	5000	2039	2954	2898	1619	1883	4	103	225	46	54	80
17	1234	20	4	4	90	5000	1943	3946	3292	1162	2054	92	5000	2051	3563	3011	1244	2364	2	108	82	-383	-281	310
18	4321	20	4	4	69	5000	1736	3365	2686	1131	1824	72	5000	1784	3220	2932	1240	1812	3	48	109	-145	246	-12
19	1234	40	4	4	85	5000	1626	4213	3012	1155	1464	86	5000	1660	3646	2972	1355	1482	1	34	200	-567	-40	18
20	4321	40	4	4	103	5000	1721	4564	3362	1247	1509	103	5000	1713	4013	3008	1552	1434	0	-8	305	-551	-354	-75
21	1234	20	8	4	151	5000	2434	5118	2812	1239	2941	158	5000	2667	5019	2725	1345	3494	7	233	106	-99	-87	553
22	4321	20	8	4	105	5000	1999	3989	2301	1199	2280	121	5000	2260	4445	2532	1311	2663	16	261	112	456	231	383
23	1234	40	8	4	134	5000	1949	5224	2594	1224	1856	149	5000	2086	4903	2830	1421	1988	15	137	197	-321	236	132
24	4321	40	8	4	165	5000	2097	5482	2983	1338	1962	170	5000	2181	5382	2747	1651	2012	5	84	313	-100	-236	50
25	1234	20	4	8	91	5000	1889	2392	3315	1156	2136	93	5000	1988	2343	3064	1248	2372	2	99	92	-49	-251	236
26	4321	20	4	8	78	5000	1736	2243	2902	1148	1887	77	5000	1770	2182	2972	1265	1870	-1	34	117	-61	70	-17
27	1234	40	4	8	88	5000	1644	2627	3157	1160	1538	90	5000	1632	2433	2937	1319	1498	2	-12	159	-194	-220	-40
28	4321	40	4	8	103	5000	1727	2874	3406	1259	1564	103	5000	1740	2491	2963	1580	1548	0	13	321	-383	-443	-16
29	1234	20	8	8	149	5000	2394	3075	2788	1235	3124	165	5000	2584	3149	2793	1331	3527	16	190	96	74	5	403
30	4321	20	8	8	132	5000	2209	2921	2590	1226	2754	136	5000	2328	2967	2700	1363	2887	4	119	137	46	110	133
31	1234	40	8	8	162	5000	2106	3480	3021	1250	2076	174	5000	2199	3433	3168	1440	2138	12	93	190	-47	147	62
32	4321	40	8	8	173	5000	2203	3441	3069	1362	2203	177	5000	2199	3311	2798	1660	2126	4	-4	298	-130	-271	-77

Exp# => Experiment Number
Rand => Randomiser seed for Mineral price variability
#Bot => Number of Miner Robots
#DBOT => Number of Digger Robots
#TBOT => Number of Transporter Robots
Time => Length of Experiment

#Mines => Total number of plots mined by the colony (Out of 320)
<All Robots and Antennas started with an initial bank balance of 1000>
Av => Average bank balance for all robots and antennas at end of experiment
D Av => Average bank balance for all Digger robots at end of experiment
T Av => Average bank balance for all Transporter robots at end of experiment
A Av => Average bank balance for all Antennas at end of experiment
B Av => Average bank balance for all Miner robots at end of experiment

Fig. 11 Test cases and corresponding productivity and profit data for "peer-to-peer" and "fixed-teams" trade models. Test cases 1 till 12 (marked red) are scenarios with bottleneck where one part of the mining process lacks resources (e.g. Availability of *Digger* or *Transporter Robots*)

9. Digger X Speed (Service Lookup table)
10. Transport X Price (Service Lookup table)
11. Transport X Speed (Service Lookup table)

In "Mine Cloud" cloud robotic system, there are more than one service providers for any particular service and a service could be demanded at several agents. All agents advertise their services and demands to the *Relation agent* that is hosting the demand or service registry (implemented as lookup tables) for their respective services and demands. A service is discovered by agents by reading the lookup tables associated with that service. Matchmaking is done based on preferences that an agent has for a particular service provider and for a particular quality and cost parameters. Preference vectors (A sequence of variables profiling an agent's service and demand preferences) of agents are matched with the quality, source and cost parameters of all available services (from different sources). This matchmaking chooses the closest matching service to an agent's preference. The matchmaking mechanism in "Mine Cloud" was hosted at the "Relation Agents" (Discovery). HTM5 however allows a designer to implement the matchmaking mechanism on any of the agents. In the absence of a matchmaking mechanism, agents could just utilize the service and demand discovery services and then negotiate bilaterally for a service contract.

Figure 9 shows the variability that exists in the simulation environment. Figure 11 shows the results and comparisons for the 32 test cases simulated on the "Mine Cloud" system. Implementation of complicated trade logics on cloud robotic colonies (Simulations and real world robots) justifies the claim of HTM5 meta-model as a usable model for development of agent oriented cloud robotic systems. As mentioned earlier in this section of the article, comparison of peer-to-peer trade methodology to any other methodology is a relatively subjective study as every methodology is suited for a particular scenario, ground rules and implementational realities. The observations presented in this section are based on the data collected in 32 test cases and for a particular trade environment. The authors do not claim that these results will stand valid for all trade environments in real world. The baseline idea is to implement complex trade logics on cloud robotic systems using HTM5 methodology, empowering its claim as a usable methodology for cloud robotic systems.

Key Observations :

1. Average profits made by *Antenna Agents* are always higher in the case of Peer-to-Peer trade. Unlike in fixed-teams scenario, the dynamic allocation of robots to a mine plot requires multiple location requests by the *Miner Robots* to the *Antenna Agents*. Service discovery and matchmaking mechanisms are hosted on a server which is connected to robots through a GSM cloud (simulated). The services of *Antenna Robots* thus are more in demand in peer-to-peer trade scenario and thus the profits made by them are always higher in the case of peer-to-peer trade.
2. Average profits made by *Agents* (Miner Bot, Digging Bot, Transporter Bot and Antenna Agents) are mostly higher in the case of Peer-to-Peer trade methodology. In 25 out of 32 cases (78.125 per cent) the peer-to-peer methodology resulted in a greater average profit as against fixed-teams trade methodology.
3. The productivity of the colony in mining the available plots is generally higher when peer-to-peer trade methodology is used. Only in 9 out of 32 cases (28.125 per cent), the productivity (Number of Mine plots processed) is lower in peer-to-peer scenario.
4. Out of the 9 cases when the productivity is lower in peer-to-peer scenario, 8 cases are from the bottleneck set (cases 1 till 12).
5. In non-bottleneck scenarios, only 1 out of 20 cases (5.0 per cent) scenarios result in a lower productivity in peer-to-peer scenario.

4 Conclusion

In this article we presented HTM5 as a feasible meta-model for designing and implementing complex trade methodologies on an agent oriented cloud robotic system. The anatomical elements of HTM5 with respect to the peer-to-peer relational trade were presented. A case study with a comparative economic analysis of peer-to-peer approach against a fixed-teams approach was presented. The primary objective of the case study was to test the feasibility of HTM5 meta-model as a design methodology for implementing complex trade methodologies on agent oriented cloud robotic systems. Design and Implementation of complex trade methodologies using HTM5 validates its claim as a feasible meta-model for agent oriented cloud robotic system development. The complete HTM5 meta-model, a domain specific language supporting HTM5 and a case study specific for peer to peer trade variability in HTM5 is currently submitted to well-known journals. Next steps in development of HTM5 are to involve industry professionals to use and improve the meta-model. A detailed user guide and a graphical design interface for HTM5 based designing is also currently under development.

Acknowledgments The current research is being funded by the Laboratory Le2i (CNRS 6306, Le-Creusot, FRANCE), Bourgogne regional council (Regional French administration) and the Universiti Technologi Petronas (Perak, Malaysia).

References

1. Dynamic Electronic Institutions, Digital Business Ecosystem and Peer to Peer Cloud Robotics Simulation Videos (2014). https://sites.google.com/site/deidbep2psimulations/
2. Alban Rousset Bndicte Herrmann CLLP (2014) A survey on parallel and distributed multi-agent systems. In: Euro-Par 2014: Parallel processing workshops, vol 8805, pp 371–382
3. Aldewereld H, Vazquez Salceda J, Dignum F, Meyer JJC (2005) Norm compliance of protocols in electronic institutions. In: Proceedings of the fourth international joint conference on Autonomous Agents and Multiagent Systems, AAMAS '05. ACM, New York, USA, pp 1291–1292
4. Arumugam R, Enti V, Bingbing L, Xiaojun W, Baskaran K, Kong FF, Kumar A, Meng KD, Kit GW (2010) DAvinCi: A cloud computing framework for service robots. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp 3084–3089. doi:10.1109/ROBOT.2010.5509469
5. Candido S., Kuffner J. (2013) Cloud-based robot grasping with the google object recognition engine. In: IEEE International Conference on Robotics and Automation, p 8
6. Dignum V (2004) A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis, Universiteit Utrecht
7. Hungate JI, Gray MM (1995) Conference report: application portability profile and open system environment users forum gaithersburg, md may 910, 1995 . J Res Natl Inst Stand Technol 100(6):699–709
8. Jennings NR, Bussmann S (2003) Agent-based control systems. why are they suited to engineering complex systems? IEEE Control Syst Mag 23(3):61–73
9. Lpez FLY (2003) Social power and norms: impact on agent behaviour. Tech. rep.
10. Luck M, McBurne P, Shehory O, Willmott S (2005) Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). AgentLink
11. Luck M, McBurney P, Preist C (2003) Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing). AgentLink
12. Muntaner-Perich E, Rosa Esteva J (2007) Using dynamic electronic institutions to enable digital business ecosystems. In: Noriega P, Vzquez-Salceda J, Boella G, Boissier O, Dignumm V, Fornara N, Matson E (eds) Coordination, organizations, institutions, and norms in agent systems II, lecture notes in computer science, vol 4386. Springer Berlin Heidelberg, pp 259–273
13. Muntaner-Perich E, Rosa Esteva J (2008) Towards a formalisation of dynamic electronic institutions. In: Sichman J, Padget J, Ossowski S, Noriega P (eds) Coordination, organizations, institutions, and norms in agent systems III, lecture notes in computer science, vol 4870. Springer Berlin Heidelberg, pp 97–109
14. Nagrath V, Meriaudeau F, Malik A, Morel O (2013) Introducing the concept of hyperactivity in multi agent systems. In: 2013 International Conference on Communication Systems and Network Technologies (CSNT), pp 542–546. doi:10.1109/CSNT.2013.117
15. Nagrath V, Morel O, Malik A, Saad N, Meriaudeau F (2013) Agent driven peer-to-peer cloud robotics. In: Cloud robotics workshop, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2013). Tokyo Big Sight, Japan. http://www.roboearth.org/iros2013
16. Nagrath V, Morel O, Malik A, Saad N, Meriaudeau F (2013) HTM5-Trade model for relationship based trade modelling in multi agent systems. In: Science and Information Conference (SAI), 2013, pp 188–196
17. Noriega P (1999) Agent mediated auctions: the fishmarket metaphor, Institut d'Investigació en Intel· ligència Artificial
18. OMG (2003) Model Driven Architecture (MDA) Guide. OMG doc. ab/2003-06-01
19. Seidewitz E (2003) What models mean. IEEE Softw 20(5):26–32. doi:10.1109/MS.2003.1231147
20. Shoham Y, Leyton-Brown K (2008) Multiagent Systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, New York, USA
21. Stone P, Veloso M (2000) Multiagent systems: A survey from a machine learning perspective. Auton Robot 8(3):345–383. doi:10.1023/A:1008942012299
22. Waibel M, Beetz M, Civera J, D'Andrea R, Elfring J, Galvez-Lopez D, Haussermann K, Janssen R, Montiel JMM, Perzylo A, Schiessle B, Tenorth M, Zweigle O, van de Molengraft R (2011) RoboEarth. Robot Autom Mag, IEEE 18(2):69–82. doi:10.1109/MRA.2011.941632
23. Wooldridge M (2002) An Introduction to Multi-agent Systems. John Wiley & Sons
24. Google goggles [latest access on 06/11/2014 0344 hrs]. http://www.google.com/mobile/goggles/
25. GostaiNet [latest access on 06/11/2014 0344 hrs]. http://www.gostai.com/
26. Internet of Things. [latest access on 06/11/2014 0344 hrs]. http://www.rfidjournal.com/articles/view?4986
27. The point cloud library (PCL) [latest access on 06/11/2014 0344 hrs]. http://www.pointclouds.org/
28. ROS the robot operating system. http://www.ros.org/
29. Rosbridge applications layer network protocol specification [latest access on 06/11/2014 0344 hrs]. http://www.rosbridge.org/
30. Rosjava [latest access on 06/11/2014 0344 hrs]. http://wiki.ros.org/rosjava
31. Web of Things. [latest access on 06/11/2014 0344 hrs]. http://www.webofthings.org/
32. TurtleBOTs (2014). http://www.turtlebot.com/
33. Visual Basic for Applications (2014). http://excelvbatutor.com
34. VisualBots Simulator (2014). http://www.visualbots.com/

Vineet Nagrath received his B.Tech. degree in computer science and engineering from the IP University, New Delhi in 2008, M.Sc. degree in computer vision and robotics from University of Burgundy (uB), France in 2011. He is currently a PHD candidate from University De Bourgogne, FRANCE and from Universiti Teknologi Petronas (UTP), MALAYSIA in a cotutelle PHD program. Since 2011, he is member of the teaching staff (software engineering, robotics and color image processing) for the Erasmus Mundus Master in Computer Vision and Robotics (VIBOT) and for MS in computer vision and robotics at uB, Le Creusot. He is a member of Centre for Intelligent Signal and Imaging Research (CISIR) at UTP. He has worked in embedded-software Industry as an engineer and has contributed to research projects in real-time image analysis with Luxscan Industries in Luxembourg. His research interests include Robotics, Software Research, Computer Vision and AI.

Dr. Olivier Morel received the M.Sc. degree in Computer vision and image processing in the University of Burgundy, in 2002. In November 2005, he received the Ph.D. degree in Computer Vision from the University of Burgundy. Since September 2007, he works as an Associate Professor in the Computer Vision team of the Le2i UMR 6306 CNRS Laboratory. His main research interests are applications of polarization imaging and catadioptric vision to robotics.

Dr. Aamir Saeed Malik is a senior lecturer in Department of Electrical and Electronic Engineering at Universiti Teknologi Petronas in, MALAYSIA. He received his BS in EE from Pakistan while MS and Ph.D. from Republic of Korea. He has more than 8 years of working experience. His research interests include 3D imaging, medical imaging and CBIR.

Dr. M. Naufal B. M. Saad obtained his Master degree in Ecole Nationale Suprieure d'Ingnieurs de Limoges (ENSIL) in France and later his PhD degree in Telecommunication in Universit? de Limoges (UNILIM), France in 2005. He is currently associate professor in the Electrical and Electronic Engineering department of Universiti Teknologi Petronas (UTP), Malaysia and also core research member in the Centre for Intelligent Signal and Imaging Research (CISIR) of UTP. His research interests cover medical imaging, intelligent signal and communication. He has authored and co-authored more than 100 international publications.

Pr. Fabrice Meriaudeau received both the master degree in physics at University of Burgundy (uB), France as well as an Engineering Degree (FIRST) in material sciences in 1994. He also obtained a Ph.D. in image processing at the same University in 1997. He was a postdoc for one year at The Oak Ridge National Laboratory. He is currently "Professeur des Universits" at uB and the Director of the Le2i (UMR CNRS). His research interests are focused on image processing for artificial vision inspection and particularly on non-conventional imaging system (UV, IR, polarization) and medical imaging. He was from 2006 to 2010 the coordinator of an Erasmus Mundus Master in the field of Computer Vision and Robotics (VIBOT). He has authored and co-authored more than 150 international publications and holds several patents in the area of non-conventional imaging systems. He was the Chairman of SPIE conference on Machine Vision Application on industrial Inspection and member of numerous technical committees of international conferences in the area of computer vision.