Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 2 (2014) pp. 101-110
© MIR Labs, www.mirlabs.net/jnic/index.html

Enhancing the Internet of Things (IoT) via the Concept of Agent of Things (AoT)

Anas M Mzahm
College of Graduate Studies
Universiti Tenaga Nasional
Kajang, Malaysia
anas1982mm@outlook.com

Mohd Sharifuddin Ahmad
College of Information Technology
Universiti Tenaga Nasional
Kajang, Malaysia
sharif@uniten.edu.my

Alicia Y. C. Tang
College of Graduate Studies
Universiti Tenaga Nasional
Kajang, Malaysia
aliciat@uniten.edu.my

Abstract—**The Internet of Things (IoT) is a relatively new concept that has many potentials and value for the research and industrial communities. It is distinguished by many features that shape the future of the Internet. However, it has some issues in intelligence, security and governance. To resolve the intelligence issue, we propose an integrated concept that constitutes the IoT and intelligent software agent technology. In this paper, we review and analyze the architectures of the IoT and identify its deficiencies, primarily the lack of reasoning and intelligence capability. We also review the software agent general architecture, the BDI architecture and multi-agent system architecture. We then propose a solution to augment the IoT with intelligent software agents resulting in a new concept called the Agents of Things (AoT). To emphasize the benefits of the AoT, we present the AoT general framework and two application scenarios of the proposed concept.**

Keywords-component; Internet of Things; Agents of Things; Multi-agent Systems; Software Agents

I. INTRODUCTION

The "Internet of Things" (IoT) is a natural evolution of digital devices in the modern era that is expected to revolutionize the interaction between the cyber world and the real world. The IoT focuses on connecting every object or things in the world to the Internet. However, this interaction suffers from many issues with regard to communication and intelligence. The ability and significance of the things in IoT to communicate with each other depends on the service type they are assigned to do [1], [2]. But communication between devices is highly programmatic and depends on the functions and schedule of interactions (i.e., reactive).

The whole interconnected system of the IoT is considered intelligent, however, the individual things are unintelligible devices. The things lack ability to reason on their environment, so that they are unable to make changes and make intelligent decisions and actions to offer other value-added services. It operates based on some implicit business rules without considering unintended changes to the environment. This is the major issue of the IoT in addition to the issues of security, governance and standardization [3], [4], [5]. All these issues could be resolved by applying the Agents of Things (AoT) concept to extend and enhance the IoT.

In this paper, we review the architectures of the IoT concept and identify its limitation and deficiencies. We also review the software agent architecture to identify the features that benefits the proposed AoT concept. We then define and propose the AoT concept and describe how the concept extends the IoT and their differences. In doing so, we propose the AoT framework and discuss issues related to the concept's design. Finally, we apply the AoT theory on two case studies of road accident monitoring and vehicle speed monitoring systems.

II. RELATED WORK

The motivation for proposing the AoT concept is to minimize the deficiencies and the limitations of the IoT concept by exploiting the benefits from the characteristics and features of the software agent technology. Consequently, in this section, we focus our review on the architecture of the IoT concept and the software agent technology. The purpose of this review will help readers to understand the motivation of proposing the AoT concept.

A. The Architecture of the Internet of Things

The ambitious vision of the IoT is to extend the Internet from the cyber world to the physical world by connecting every object to the Internet [1], [2]. The things in this concept vary from physical objects to cyber entities, such as television, computing devices, software entities, etc. [6], [7]. The IoT concept enables these things to connect and communicate with each other or to be controlled remotely. This creates an environment for sharing information between the things in real-time. The environment unites the physical world and the virtual world together [3], [8], [9], and creates a link to exchange data between real devices and cyber applications in a secured connection [10].

The IoT concept is distinguished by its dynamic architecture, due to some characteristics that enable it to share information, intelligent handling and large-scale interaction. Sharing information represents the functions of getting and exchanging information with things in one hand and with other devices over the Internet in another, while intelligent handling represents the ability of processing and controlling information intelligently. The final notable characteristic of IoT is the large-scale characteristic of connected things. Some researchers estimate the number of connected things could soar from 90 billion to nearly 600 billion things [9], [11], [12], [13]. The simplicity of the IoT architecture is another aspect that makes the IoT architecture dynamic. The three-hierarchical layer architecture is formed by the perception layer, transport layer and application layer [14], as shown in Figure 1.

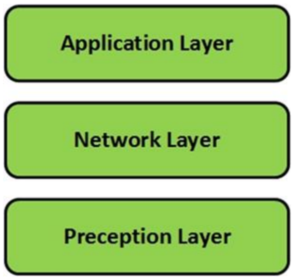


Figure 1. The Three-layer Architecture of IoT

The working principle of the hierarchical architecture starts when the Perception layer collects data of connected things through its sensing technology, such as RFID and Sensors. Then, it transfers the collected data to the next layer which is the Transport layer. This layer uses the communication methods of the Internet or local network to carry the collected data to the applications in the Application layer for processing. Finally, in this layer the data is processed and analyzed to be stored in databases or to be shared with other application systems [15].

B. Intelligent Software Agents Architecture

For a long time, Intelligent Software Agent is an interesting research field and seen as a technology that has many potential to solve a variety of issues. The paradigm has its notable application in distributed systems. Nowadays, with every achievement and milestone in technology, software agents represent a leading solution to solve issues related to complexity and diversity of modern systems [16], [17]. Wooldridge [18] defines a software agent as a computer system that is able to interact with its environment and capable of making autonomous decision on behalf of its owner to meet its given objectives (see Figure 2). Another definition for software agent is a software entity that works autonomously and continuously in a specific environment inhabited by other software entities [17], [19].

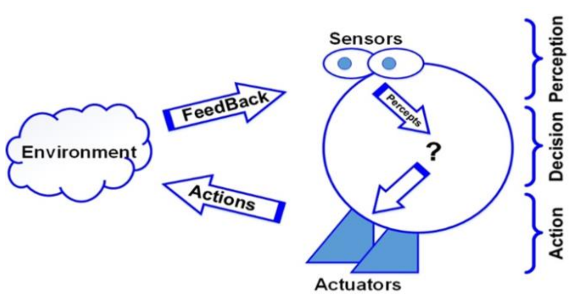


Figure 2. Simple representation of software agent and environment

From these definitions, we can derive some distinct properties about software agents. Firstly, software agents are notable for being flexible and intelligent, so that it can cope with any environmental changes and respond to it without asking, to some extent, for its owner's interference and guidance [20], [21]. In addition, software agents have the authority to do what they see suitable in any way to achieve their objectives and goals [18]. Therefore, to achieve the given goals, software agents work autonomously and continuously in their environment, that enables them to learn from experience and form some kind of knowledge base. Finally, software agents are not alone in an environment, thus entailing their ability to interact, negotiate and cooperate with other agents and software entities in that environment [20]. All these properties can be summarized to three distinguished characteristics: Reactivity, Pro-activeness and Social ability [18], [20], [22]:

- Reactivity: Software agents have the ability to sense the surrounding environment and interact with it in manners that serve their objectives.
- Pro-activeness: Software agents have the ability to change their behavior to be goal-directed by starting the first request or contact (take the initiative) in order to achieve their objectives and goals.
- Social ability: Software agents have the ability to be social and interact with other agents to achieve their objectives.

Software agents enjoy many other characteristics, which include mobility, interactive, adaptive, coordinative, cooperative, negotiation, etc. [18], [23], [24]. Software agents enjoy a wide range of other characteristics, making it impossible for any researchers to include all of these in a single type. At BT labs, they reduce these characteristics to three important attributes only, which any basic software agent architecture could have as shown in Figure 3 [20], [24].

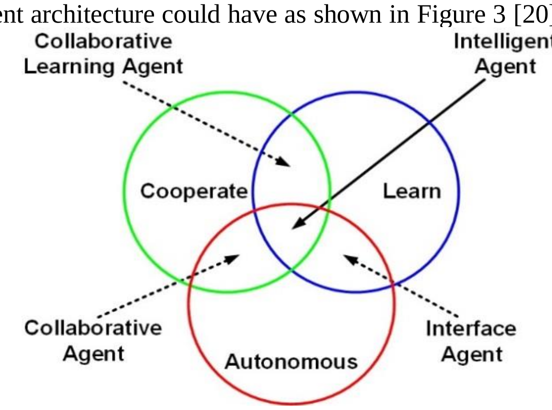


Figure 3. Software agents general architecture

From these attributes they derived many types of software agents, which are divided into seven main types as shown in Figure 4 [24]:

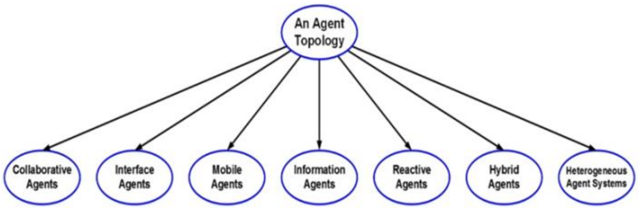


Figure 4. Software Agents Types

- Collaborative Agents

This type of software agent is characterized by its autonomy and cooperation with other agents to achieve the tasks of its owners. In addition, in order to achieve its delegated goal, this agent type is able to negotiate with other agents to reach an acceptable agreement. The characteristics of collaborative software agents include autonomy, social ability, responsiveness and pro-activeness [24].

- Interface Agents

This type of software agent is characterized by its autonomy, learning capability and working with users in the same work environment. An interface software agent represents a personal agent (autonomous personal assistant) that helps its owner by observing, monitoring and learning the actions. It then suggests new methods and better ways for doing the actions in the application. The agent, while collaborating with the user may not need to use an agent communication language as with other agents. Moreover, its collaboration with other agents is limited to asking for advice only and not for negotiation like collaborating agents [24].

- Mobile Agents

A mobile agent is a software agent that is characterized by mobile capability, i.e. it is capable of transporting itself from one location to another. In addition, it is autonomous, cooperative and capable of travelling through computer networks to interact with foreign hosts and gather information on behalf of its owner and then returns to its owner after performing its delegated tasks. This type of software agent has a unique feature represented by its capability of exchanging information with other agents without giving all its information [23], [24].

- Information Agents

The main purpose for information agents is to help its owner to manage, manipulate and collect information from many distributed resources. It is identified by what it does unlike collaborative and interface agents. It is also characterized by its autonomous actions and mobility [24].

- Reactive Agents

Reactive agents are featured by their simplicity and basic interaction with other agents. The agent is visioned as a collection of modules that function autonomously to perform a particular task. It also characterized by dynamic interaction with its environment, which leads to undesired complexity. Reactive agents are categorized as a low-level nature that is close to raw sensor data [21], [24].

- Hybrid and Heterogeneous Agents

Hybrid agents are formed by combining two or more type of software agents in one single entity. This type of agents is used for an improved version of the software agent, which has the strength of the combined types, to meet the need of the designer's goals. Heterogeneous agents are similar to hybrid agents and are formed for the same purpose of improving the strong points or to reduce the weak points in the combined types. However, heterogeneous agents may include hybrid agents as well [21], [24].

C. *Multi-agent System Architecture*

The software agent technology manifests a new paradigm in resolving many common problems that could be solved by structured and object-oriented paradigms. However, for large scale, complex distributed systems, existing programming paradigms becomes unwieldy and complicated [24], [25]. Researchers in Artificial Intelligence (AI), particularly those in software agent community, have attempted to resolve these complex issues by using groups of software agents working together by imitating the behavior of humans' societies and applying the concepts on software agents [17], [24], [26], [27]. The core idea is to build an agent with adequate knowledge and group several agents in an environment to create a system where each agent is assigned to a task. The concept manifests what is termed as a multi-agent system. Multi-Agent Systems (MAS) have become one of the most important research field in AI [28].

A multi-agent system contains a number of independent, autonomous software agents and capable of interacting and communicating with each other by exchanging messages. These agents cooperate, coordinate and negotiate with each other to achieve the goals delegated by its owners [17], [18]. Every agent in the system works toward achieving its sub-goals, even if these sub-goals contradict with other agents (i.e. they either cooperate or compete).

MAS systems are self-organizing systems and usually work without the interference of its owners. The system works by breaking down a complex task into subtasks, then assigning each subtask to an agent to work on based on its knowledge base. If an agent has no knowledge to do the subtask, the agent is able to ask other agents in the system for advice or delegates the job to them. The agents in these systems work together as one entity or they work individually, which represents several separate entities [17], [29]. A society of agents may include several many sub-societies of agents [30]. Figure 5 show an example of a MAS system called Genie of the net: A simple multi-agent system [31].

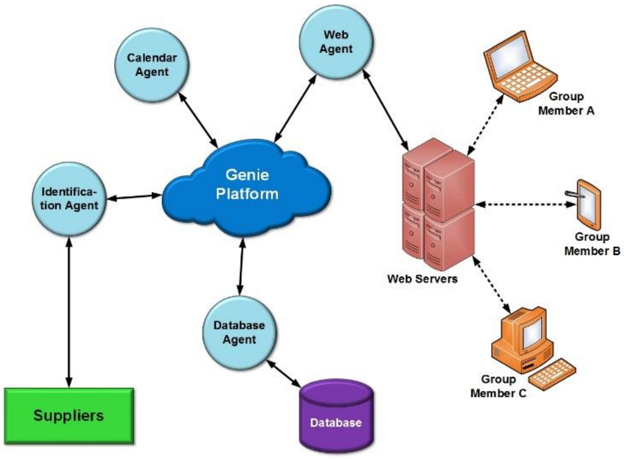


Figure 5. A simple MAS system

D. *BDI Agent Architecture*

The belief-desire-intention (BDI) architecture represents one of the most durable software agent architecture in existence [32]. This architecture is developed by Stanford Research Institute under the name of Procedural Reasoning System (PRS). The PRS and the BDI are mutually used to refer to the BDI software agents. The data structure representation in this architecture is consistent with the mental states of the BDI paradigm [33]. The BDI agent architecture is shown in Figure 6.

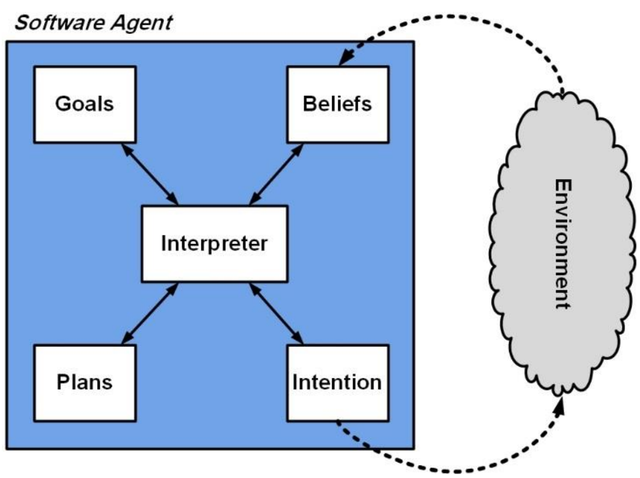


Figure 6. The PRS or BDI agent architecture

The working principle of the BDI agent starts when the agent attempts to achieve a given goal in the desire stack. The first thing the agent does is to push the selected goal from the desire stack to the intention stack, where all ongoing goals are placed. It then searches for a plan from the plan library to enable it to achieve the goal. The plan library contains a set of plans that the agent could use to achieve different goals. The agent chooses a suitable plan and executes it to achieve the goal based on its beliefs about the prevailing conditions. During this phase, the agent may add more goals to the intention stack and pursue to achieve them. However, if one plan fails to achieve the goal, the agent is able to choose another plan from the plan library [18].

III. DEFICIENCIES OF THE IOT

The vision of the IoT concept is to establish a large-scale system [15] by connecting every thing in the world to the Internet and making them communicate with each other, machine-to-machine (M2M). The IoT attempts to assist humans in deciding better data management solutions for their domain-specific problems by processing relevant information from interconnected things. It extends the current Internet-based technology by connecting different types of things (objects or devices) with each other and enable them to be communicated smartly [15]. Consequently, this concept is designed to connect millions of things together, but things of this number need large storage spaces and generate heavy traffic, which could create many network issues. Furthermore, while the things are connected with each other, they are not necessarily able to communicate with each other [3]. Their ability to communicate with each other depends on the similarity of the service they are assigned to do [1]. These deficiencies are due to the lack of ability of the things to reason on their environments and subsequently make intelligent decisions and actions to achieve their objectives. While the entire connected system of the IoT manifests the ability to make decisions and interact smartly between the things, it does so based on some implicit business rules without considering unintended changes to the environment. The whole system can somewhat be represented as an intelligent system but not the things. These are the major issues of the IoT in addition to the issues of security, governance and standardization [3]. These deficiencies somewhat delays the vision of the IoT concept from becoming a reality.

A summary of these deficiencies is as follows:

- A single thing in the IoT is not smart but the whole system is somewhat a smart system.
- The decision making and the subsequent actions are service-dependent.
- The things in the IoT are connected with each other but they are not necessarily able to communicate with each other.
- The things are able to communicate with other things only when they are designed to perform the same system services.
- The IoT suffers from other issues regarding security, governance and standardization.

IV. THE AGENTS OF THINGS (AOT) CONCEPT

The Agents of Things (AoT) concept is proposed to enhance the Internet of Things (IoT) concept [34]. The main idea of AoT is that every thing in this concept should have an optimal internal reasoning and intelligent capabilities. These capabilities enable the things to interact directly with other things in the same or different system types. The reasoning and intelligent capabilities in these things are handled by software agents. Therefore, all the things in this concept see and interact with each other as an intelligent system. This feature enables the things to be independent and mobile if needed. We define the AoT concept as follows:

"The Agents of Things is a concept that extends the Internet of Things by embedding things with intelligent software agents to give them the power of self-reasoning and intelligence in providing value-added services to humans".

A. How AoT Expands IoT

Implementing AoT is not as trivial as connecting those things in an Internet-like structure and establishing Internet communication protocols. Important and significant issues in manifesting the AoT include the need for agents to reason on its environment, identify specific actions to take and whether those actions are within its autonomy. The intelligence of these things needs optimized resources to run the software agent program.

The intelligence in these things is represented by software agents embedded in the things and their ability to reason about their environments in contributing useful outcomes for humans. Therefore, all these things and systems that work under the AoT concept interact with each other via software agents. Figure 7 shows the proposed architecture of the AoT in software agent's perspective.

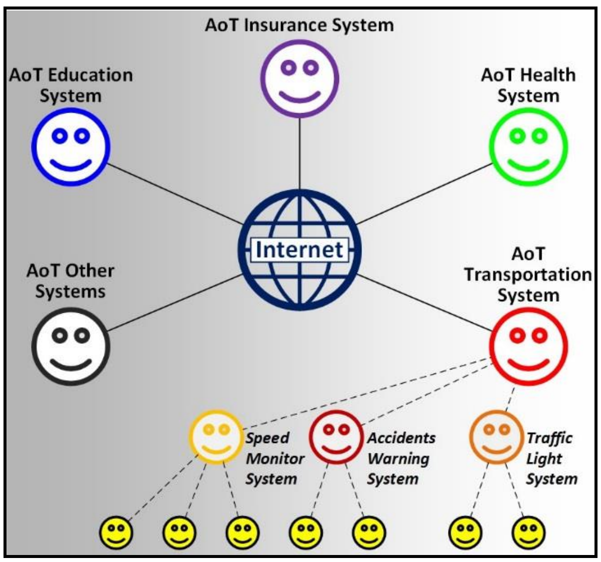


Figure 7. The General Architecture of the AoT

B. Differences between AoT and IoT

The difference between AoT and IoT is that the AoT is an intelligent concept that uses software agents to give its associated things the ability of reasoning, negotiation and delegation. However, in the IoT concept, the things are not intelligent since it does not include any artificial intelligence software to give them the reasoning ability. However, the whole system of the IoT concept is somewhat intelligent due to its collective behavior. The ability of things to communicate with each other over the Internet gives them the characteristic of intelligence because their operations are smarter than the things that are not connected to the Internet [35].

C. The AoT Framework

It is a challenge to design a framework for the concept of AoT without considering the software and hardware perspectives of this concept. This concept merges the software perspective, which is represented by software agents to implement the functional requirements of the things and the hardware perspective, which provides the platform for software agents. In the software perspective, the things see each other as software agents and communicate with each other based on this notion. In the hardware perspective, these things are connected to each other in their original hardware device configuration, which sends and receives signals from each other.

We propose an AoT framework in Figure 8, which shows that the framework constitute two perspectives: software and hardware. The software perspective contains the software form of things, which is represented by software agents. These agents communicate with the application systems through exchanges of data, information, messages, requests, etc. The hardware perspective contains the physical parts of things, through which connection is established with the Internet and other things. These two perspectives manifest intelligent interactions between the agents of these things.

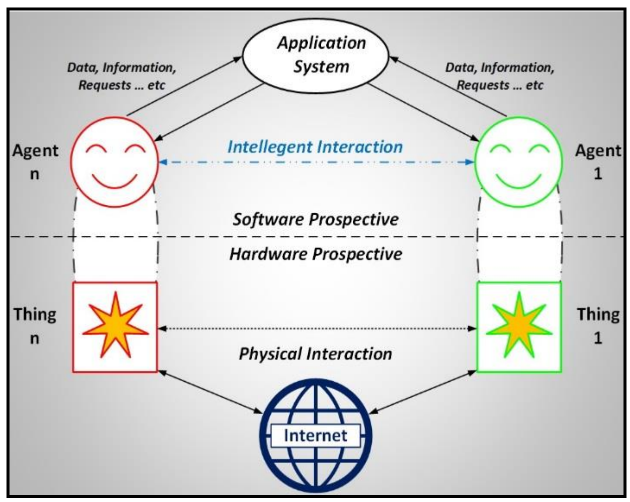


Figure 8. The Agent of Things framework

D. Design Issues in AoT

There are some issues in the AoT concept regarding the communication, reasoning level, and agent capabilities that form the basis for this research. These issues are:

- The optimal agent actions that are required for the things' functions.
- The optimum reasoning level of the agent embedded in the things.
- The agent's ability to evaluate the conditions of the environment and produce the proper actions in providing value-added services to human beings.
- The optimum communication capability for the agent.

- The agent's capability to communicate with other agents via some communication protocol. For example, an intelligent refrigerator could remind its owner to buy extra foodstuffs, which are depleting or communicate with the suppliers system to order more foodstuffs. In this way, services are improved, increasing the productivity and efficiency of work when things remind, inform, or suggest actions that could reduce time, costs, and wastes.
- The optimum architecture of things required to embed an agent that performs the things' functions.
- Since things are of many different sizes and functions, an optimum level of autonomous agent-based functions must be determined to be embedded in those things.
- Matching the things architecture to that of software agent architecture.

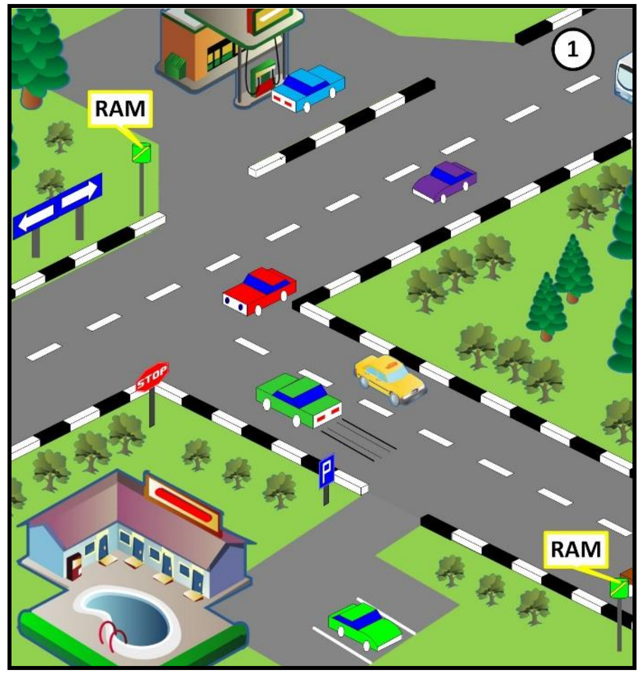
V. Modeling AoT for Traffic Systems

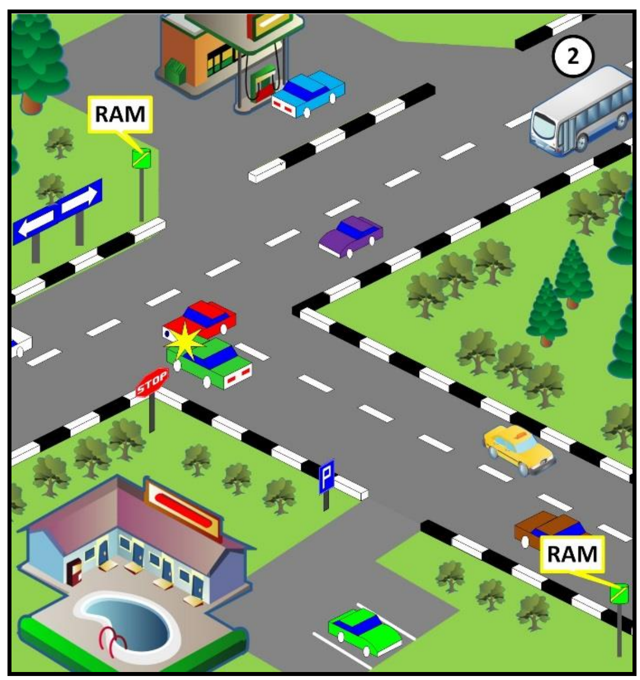
The car or automobile is a remarkable machine that has revolutionized the transportation industry since its invention. It greatly changes our lives and makes it easier for us to reach distant places in a very short time, as we work in the city while we live in the suburbs of the city. We become more dependent on the car to use it as our main transportation mode. Technological advancement makes the car lighter and more powerful to run at higher speed. This combination transforms the car from a convenient transportation vehicle to a deadly machine that can endanger our lives and the lives of others. Indeed, road accidents remain one of the top killers in many countries. Consequently, vehicle speed monitoring is one of the most important operations of the traffic authorities.

A. Road Accident Monitor

It is a fact that road accidents happen every day around the world. Whatever the reasons for these accidents, many people lost their lives every day. The situation on the road lacks a rapid accident alerting and warning system that can be used to alert authorities for rescue services and warn other drivers about accidents. We propose to mitigate this problem by applying the AoT concept as shown in Figure 9.

In the figure, every road is monitored by a device, which we called the "Road Accident Monitor" (RAM). This device is equipped with software agent-based functions for handling road accidents. When a road accident happens, (see sequence of Frames 1, 2, 3 and 4 of Figure 9) the cars that are involved in the accident inform the RAM through their agents with a message, e.g., "I have an accident, please help". The RAM responds by taking a series of steps based on the functions accorded to its software agent.





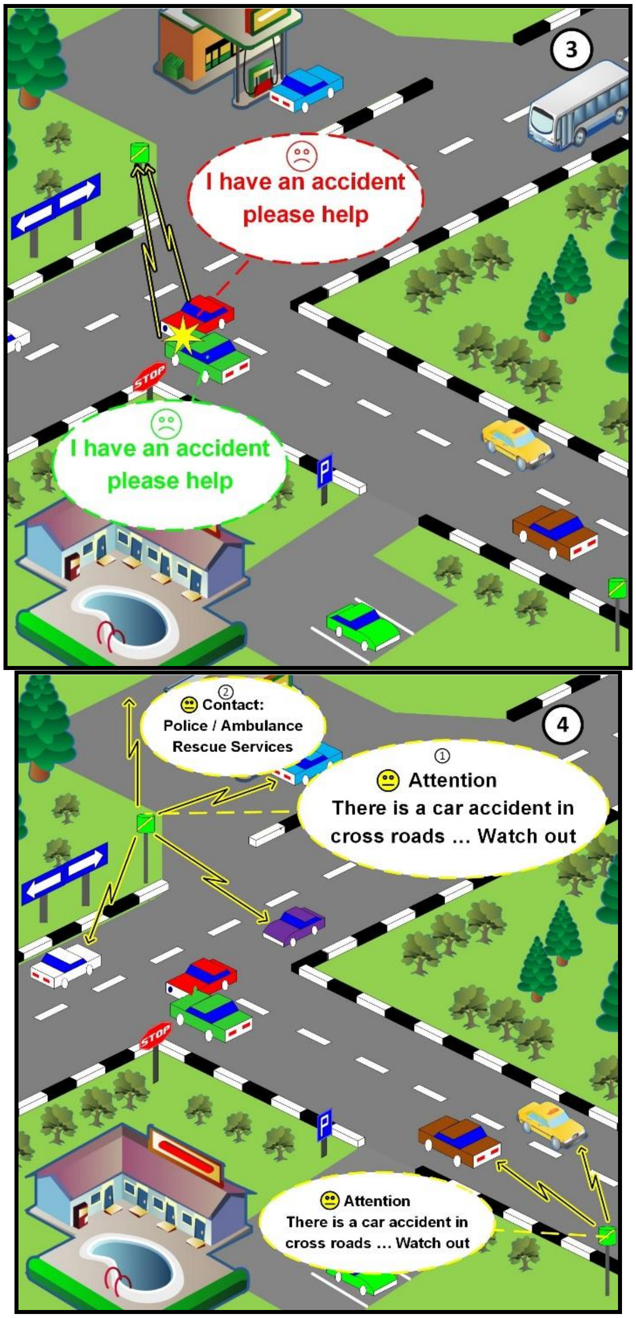


Figure 9. RAM in Operation

Firstly, the RAM continuously warns other vehicles approaching the scene of the accident ahead. Secondly, it informs the nearest police unit and emergency service about the accident at the location and request immediate assistance. Finally, if there is a need, it starts a private channel for these cars that are involved in the accident to communicate with other services, such as insurance, and next of kin. The beneficial outcome of such message exchanges from this AoT system is a fast and rapid response by the authorities and trauma services to immediately attend to the victims of the accident.

The Road Accident Monitor system can be improved further by designing the system as an external device that monitors the road and communicate with vehicles. In the near future, when vehicles have the ability to connect to the Internet as a standard option, every vehicle on the road can communicate with other vehicles and pass warning messages between them and with the authority. This improvement in the AoT resembles the undergoing research by Mercedes-Benz called Car-to-X technology [36] and Open Robo Car by University of Michigan [37].

B. *Vehicle Speed Monitoring*

The main reason for limiting vehicle speed on certain roads is to prevent road accidents. To check on drivers from travelling over the speed limit, traffic police uses various methods. We review the evolution of these methods and analyze their advantages and disadvantages. We then propose our method for detecting speeding vehicles on the road based on the AoT concept.

Two methods have been used for detecting speeding vehicles on the road. In the first method, a police officer sets a speed trap at some part of the road where he/she holds a portable Doppler radar pointed at oncoming vehicles to measure their actual speed (see Figure 10, Frames 1 and 2). When a speeding vehicle passes the officer with the radar, he/she detects its speed. If it exceeds the speed limit, he/she then informs his/her counterpart over the police radio to stop the offending vehicle at a roadblock further up the road.

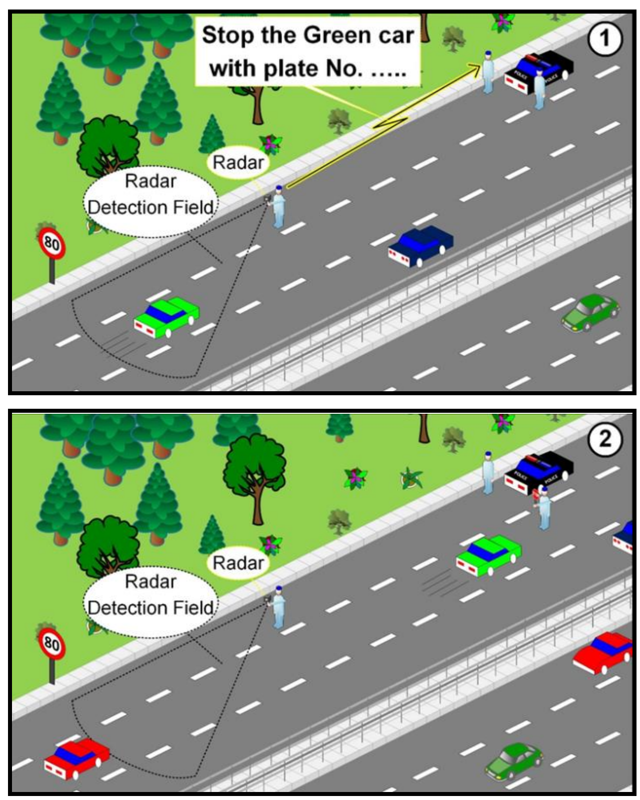


Figure 10. First generation of speed detection method.

The drawback of this method is that it cannot be applied on every road because of resource limitations especially the police force. In addition, the police force can only work for a limited number of hours. The weather conditions further influence the motivation of carrying out the operation.

In the second method, speed cameras are used to ‘catch’ the offending vehicle. These speed cameras have built-in radar to detect vehicles’ speeds. When the camera detects a speeding vehicle, it takes a picture of the car registration number, as shown in Figure 11 (Frames 1 and 2). It then sends the image to the police’s vehicle administration system to analyze the image and issue fine to the car owner.

There are many reasons that render these cameras inefficient and annoying to most drivers. Firstly, drivers can evade detection by these cameras by memorizing their locations, slowing down their vehicles when they reach the detection point and speeding up again when they pass it. Secondly, some drivers use fake number plates to fool the cameras with false information of the offenders in captured images. Finally, it is annoying for some drivers who do not know the road or who are tired from long distance driving and missed the warning sign of cameras. All these issues are drawbacks of speed cameras. On the contrary, speed cameras are deployable twenty-four hours a day despite the weather conditions and the fines reach the drivers anywhere.

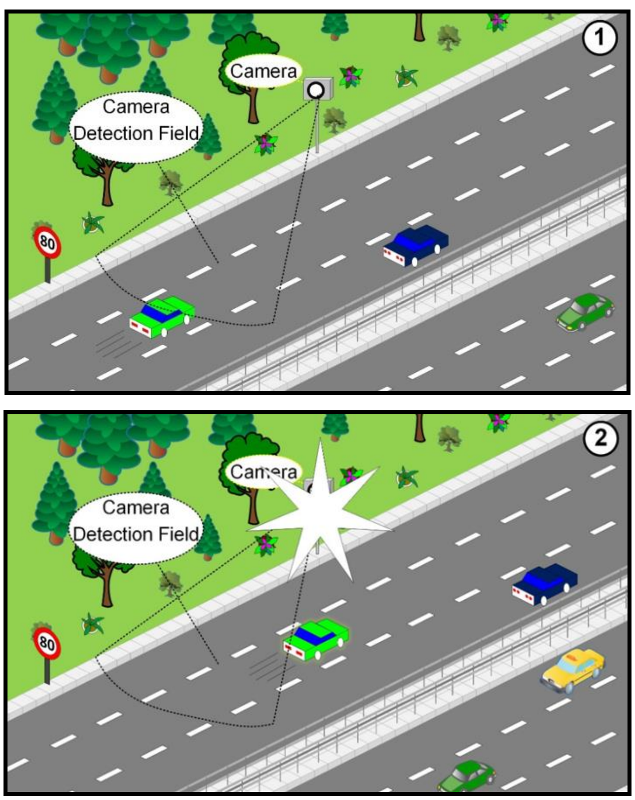
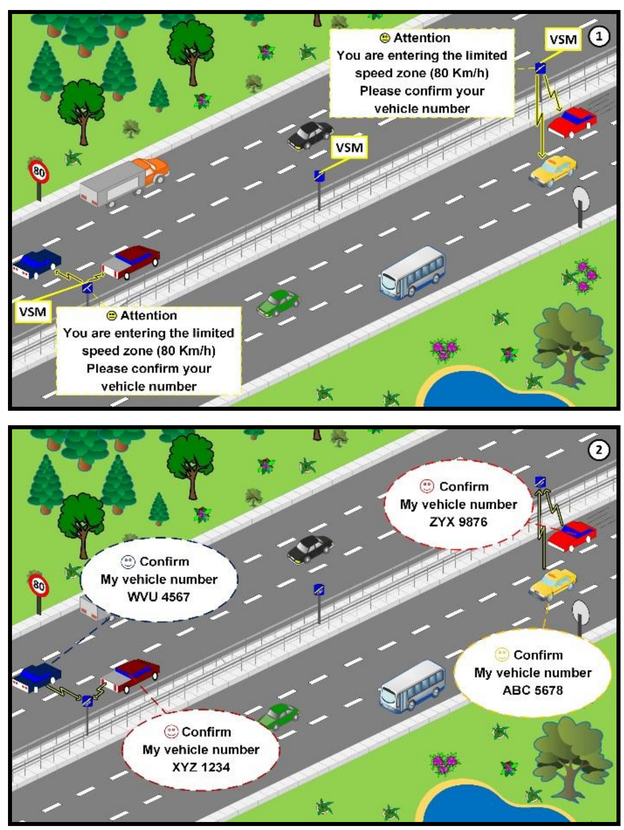


Figure 11. Second generation of speed detection method.

The drawbacks and problems in the current speed monitoring methods could be resolved by applying the AoT concept. Using this concept, we propose a “Vehicle Speed Monitoring” (VSM) system to monitor speeding vehicles on the road. In this scenario, every vehicle is fitted with a software agent that carries the vehicle’s and owner’s details. The VSM devices include a software agent to monitor and cover the whole road. It could be fitted on street lamps, roadside buildings or purpose-built structures to detect vehicle speed on the road as shown in Figure 12 (Frames 1, 2, 3, and 4).

The software agent of the VSM device contacts each vehicle on the road and informs the driver that he/she has entered a speed limit zone. The initial contact is considered as a cautionary message to inform the driver and to draw his/her attention to the situation. If he/she exceeds the speed limit, his/her details are alerted by the agent to the authority for further actions.

All vehicles and their details (driver’s name, vehicle registration number, etc.) are monitored at all times while they are in the speed limit zone. The VSM requests each vehicle’s agent to confirm the vehicle’s details, which are kept by the agent. This operation is necessary to confirm the identity of the vehicles and their drivers for subsequent data processing functions, e.g. issuing speeding ticket or final warning to drivers. The benefits of this AoT application are efficient identification and fast action against offenders. We consider the VSM system as a third generation method of detecting and monitoring vehicle speed.



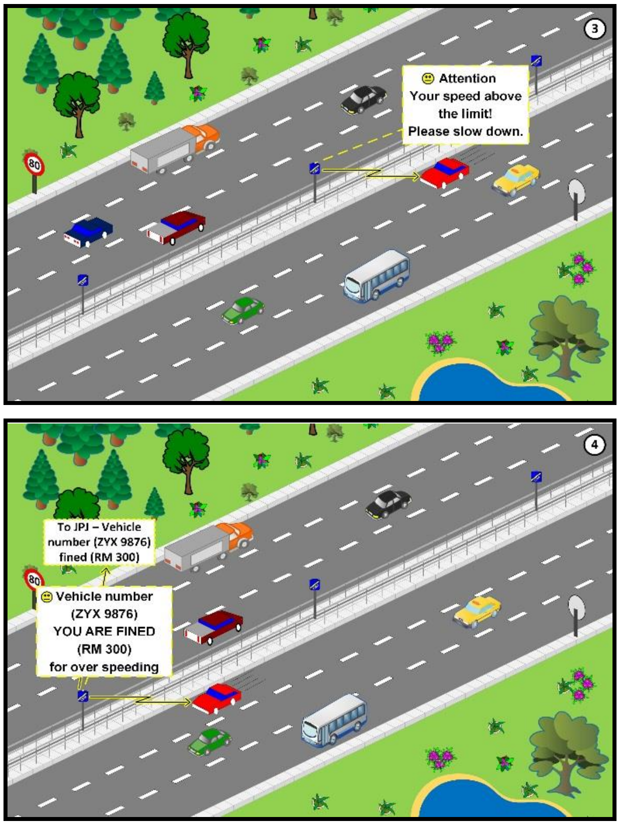


Figure 12. The AoT Speed Monitoring Scenario

The intelligence that distinguishes the AoT concept from other systems is the reasoning and communication abilities of the software agents to react to the dynamism of the environment. In these scenarios, the RAM/VSM devices and the vehicles are things and objects in the environment that interact via the software agents in the things.

VI. DISCUSSION

Many factors could affect the flow of the traffic system and threaten the lives of its users. Factors such as high-speed driving, careless drivers, bad weather, vehicle malfunction, etc., could be a cause of traffic accidents. Many of these accidents could be prevented and human casualties could be minimized, if there is an effective warning system for accidents. Such a system can monitor the roads for accidents and warn other vehicles and the authority of any accidents on the road.

In our research, we propose a warning system that could save human lives and improve the authority's response time during any accident. Using the AoT-based concept, the RAM system warns other drivers of any accident ahead, communicates with government authorities and provides other necessary services. While the RAM is an important system, it is inadequate to minimize losses of human lives. Consequently, we expand the AoT concept to include the factors that could cause the accidents in the first place, such as high speed driving.

High speed driving is thought to be the major cause of traffic accidents. Authorities around the world monitor traffic speed on the roads with two different generations of methods. However, these two speed detection methods can be influenced by several factors, such as human error, bad weather, devices malfunction, falsified information and lack for proper warning notice. Consequently, we propose our AoT-based concept, the VSM system, as a third generation high-speed detection method. This system minimizes the influential factors and provides proper monitoring and warning notice at all times. The system enforces the vehicle drivers to maintain the speed limit at all times and this somewhat reduces traffic accidents and the subsequent human casualties.

The implications of applying the AoT concept in traffic systems include saving human lives to saving time by minimizing the time taken to react on the road. The benefits of continuously monitoring vehicles' speed on the road help to save lives. This could indirectly influence drivers' temperament and prevent them from accidents. The benefits of saving human lives and better control on traffic flow are reflected directly on the authority and government. They will optimize resources (police force, ambulance, rescue services, etc.) to solve problems and accidents on the roads.

While the Internet of Things is revolutionizing the way things are managed, it lacks the intelligence in its architecture to capitalize on changes of the environments that offer value-added services to humans [34]. The Agents of Things extends and enhances the IoT concept by embedding the things with intelligent software agents, which enables the things to reason on its environment. The important issues in IoT that the AoT concept would resolve are the communication constraint and service type dependency. The AoT would enable the things to communicate freely with other things despite their services' and systems' types. The AoT also enables the things to have self-reasoning capabilities, by sensing their environments and interacting with other things based on current situations. The benefits of the AoT are beyond our imagination.

VII. CONCLUSION AND FURTHER WORK

In this paper, we review the architecture of the IoT and software agents, which we use to conceive the AoT concept. We discuss how AoT extends IoT and the benefits of using the AoT concept to solve real world issues, such as road accident monitoring system and vehicle speed monitoring system.

In our future work, we shall proceed to investigate the optimum reasoning capabilities that should be given to things that correspond to its size and function. This can be done by analyzing the available computing hardware in the market that are used as things. Correspondingly, we shall also analyze the software agent communication methods, domain actions, domain knowledge and reasoning capabilities. The result from these analyses will help us to match the hardware devices and software agent functions to find the optimum reasoning level for each category of things.

REFERENCES

[1] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key

challenges," in Proceedings of the 2012 10th International Conference on Frontiers of Information Technology, 2012, pp. 257-260.

[2] N. Bari, G. Mani, and S. Berkovich, "Internet of things as a methodological concept," in Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on, 2013, pp. 48-55.

[3] L. Tan and N. Wang, "Future internet: The internet of things," in Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, 2010, pp. V5-376-V5-380.

[4] I. Friese, J. Heuer, and K. Ning, "Challenges from the Identities of Things: Introduction of the Identities of Things discussion group within Kantara initiative," in Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 1-4.

[5] A. F. Skarmeta, J. L. Hernandez-Ramos, and M. Moreno, "A decentralized approach for security and privacy challenges in the Internet of Things," in Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 67-72.

[6] Y. Kang and Z. Zhongyi, "Summarize on Internet of Things and exploration into technical system framework," in Robotics and Applications (ISRA), 2012 IEEE Symposium on, 2012, pp. 653-656.

[7] K. Sye Loong, S. S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A Standardization Perspective," Internet of Things Journal, IEEE, vol. 1, pp. 265-275, 2014.

[8] Y. Huang and G. Li, "A semantic analysis for internet of things," in Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on, 2010, pp. 336-339.

[9] P. Leong and L. Liming, "Multiagent Web for the Internet of Things," in Information Science and Applications (ICISA), 2014 International Conference on, 2014, pp. 1-4.

[10] T. Fan and Y. Chen, "A scheme of data management in the Internet of Things," in Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on, 2010, pp. 110-114.

[11] P. Biggs, L. Srivastava, and I. T. Union, ITU Internet Reports: The Internet of Things: International Telecommunication Union, 2005.

[12] S. E. Sarma, "Towards the five-cent tag," Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. Available from http://www. autoidcenter. org2001.

[13] H. Zuerner, "The Internet of Things as greenfield model: A categorization attempt for labeling smart devices," in Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 5-9.

[14] G. Gang, L. Zeyong, and J. Jun, "Internet of things security analysis," in Internet Technology and Applications (iTAP), 2011 International Conference on, 2011, pp. 1-4.

[15] C. Xiang and X. Li, "General analysis on architecture and key technologies about Internet of Things," in Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on, 2012, pp. 325-328.

[16] F. Zambonelli and A. Omicini, "Challenges and research directions in agent-oriented software engineering," Autonomous Agents and Multi-Agent Systems, vol. 9, pp. 253-283, 2004.

[17] M. H. Laarabi, C. Roncoli, R. Sacile, A. Boulmakoul, and E. Garbolino, "An overview of a multiagent-based simulation system for dynamic management of risk related to Dangerous Goods Transport," in Systems Conference (SysCon), 2013 IEEE International, 2013, pp. 830-835.

[18] M. Wooldridge, An Introduction to Multiagent Systems: John Wiley & Sons, 2009.

[19] Y. Shoham and M. Tennenholtz, "On the emergence of social conventions: modeling, analysis, and simulations," Artificial Intelligence, vol. 94, pp. 139-166, 1997.

[20] J. M. Bradshaw, Software agents: MIT press, 1997.

[21] A. Leite, R. Girardi, and P. Novais, "Using Ontologies in Hybrid Software Agent Architectures," in Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, 2013, pp. 155-158.

[22] V. S. Lazarou, S. K. Gardikiotis, and N. Malevris, "Agent Systems in Software Engineering", 2008.

[23] H. Razouki and A. Hair, "Components-based software architecture for secure mobile agents via two strategies of adaptation," in Intelligent Systems: Theories and Applications (SITA), 2013 8th International Conference on, 2013, pp. 1-7.

[24] H. S. Nwana, "Software agents: An overview," The knowledge engineering review, vol. 11, pp. 205-244, 1996.

[25] P. Van Roy, "Programming paradigms for dummies: What every programmer should know," New computational paradigms for computer music, vol. 104, 2009.

[26] J. Ferber, "Simulating with reactive agents," Many Agent Simulation and Artificial Life, vol. 36, pp. 8-28, 1994.

[27] H. Nwana, "Simulating a Children's Playground in ABLE," Working Report, 1993.

[28] A. A. Hopgood, Intelligent systems for engineers and scientists: CRC press, 2011.

[29] G. Elofson, P. M. Beranek, and P. Thomas, "An intelligent agent community approach to knowledge sharing," Decision support systems, vol. 20, pp. 83-98, 1997.

[30] M. N. Huhns, "Agents as Web services," IEEE Internet Computing, vol. 6, p. 93, 2002.

[31] J. Riekki, J. Huhtinen, P. Ala-Siuru, P. Alahuhta, J. Kaartinen, and J. Röning, "Genie of the net, an agent platform for managing services on behalf of the user," Computer Communications, vol. 26, pp. 1188-1198, 2003.

[32] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, "The belief-desire-intention model of agency," in Intelligent Agents V: Agents Theories, Architectures, and Languages, ed: Springer, 1999, pp. 1-10.

[33] M. J. Wooldridge, Reasoning about rational agents: MIT press, 2000.

[34] A. M. Mzahm, M. S. Ahmad, Alicia Y. C. Tang, Agents of Things(AoT): An Intelligent Operational Concept of the Internet of Things (IoT), 13th International Conference on Intelligent Systems Design and Applications (ISDA 2013), pp. 159 – 164, 2013.

[35] E. Fleisch, "What is the internet of things? An economic perspective," Economics, Management, and Financial Markets, pp. 125-157, 2010.

[36] Mercedes-Benz. (2014, 20 June 2014). Car-to-X Communication. Available: http://ces.mercedes-pressevents.com/content_drivestyle.php

[37] W. Jones. (2014, 20 Jun 2014). University of Michigan to Open Robo Car Urban Test Track in the Fall (10 Jun 2014 ed.). Available: http://spectrum.ieee.org/cars-that-think/transportation/self-driving/university-of-michigan-to-open-robo-car-test-track-in-the-fall