Concepts for Conceptual Modelling of an IoT Application

Naveen Prakash[1] and Deepika Prakash[2]
[1]*Institute of Computer Learning and Consultancy, 21/4 S.B.S. Marg, New Delhi 110001, India*
[2]*Department of Computer Science, NIIT University, Neemrana, 301705, India*

Keywords: IoT-Application Design, Information System of Things, Agent, Communication, Aspects.

Abstract: The design of IoT applications has been considered as a specification of devices and their inter-connection. In terms of the Systems Development Life Cycle, this addresses the stage of implementation design. Without explicitly modelling the application problem, this could lead to ill-fitting devices, missing processing of data and inappropriate communication choices. To obviate this, the upstream stage of conceptual modelling is explored in this paper. By, analogy with Information Systems, IoT applications are looked upon as Information Systems of Things, ISoT and the conceptual model for an ISoT is referred to as the Conceptual Model of Things, CMoT. It is shown that agent-orientation is more appropriate to ISoT than object-orientation. Agents have aspects that are its measurable properties. The proposals are illustrated with an example.

1 INTRODUCTION

The internet of things, IoT, is a collection of objects like sensors, cameras etc. that communicate over the internet (Keoh et al., 2014). Thus, the IoT is largely about interaction (Yanwei et al., 2011) between things (by sending messages, signals to each other) and has an interface to human beings. According to Oracle (https://www.oracle.com/in/internet-of-things/what-is-iot/), "The internet of things describes the network of physical objects …. for the purpose of connecting and exchanging data with other devices and systems over the internet". The internet of things has been looked upon as organized in layers, for example,

- Device Layer: To enable co-operation and integration, ontologies for the objects comprising the IoT have been developed. Thus, we have sensor ontologies like SSN (www.w3.org/2005/Incubator/ssn/) of W3C, and actuator ontologies like SAN (www.irit.fr/recherches/MELODI/ontologies/SAN). Additionally, in (Hachem et al., 2011), there are three levels of ontologies, device, domain, and estimation levels. Wang et al (Wang et al., 2012) propose a single level ontology for IoT services having seven concepts.
- Sensor Network Layer: Each sensor network consists of a large number of sensing nodes. There is a special node called sink that is used to collect sensing results reported by other nodes in the network. Sensor networks cooperate, for example, with RFID systems to enable object tracking.

A five-layer architecture can be found in (Zhong et al., 2015) and a three-layer architecture in (Mahmoud et al., 2015).

The term IoT system has been used in two senses. In the first sense, it refers to a collection of IoT things together with the middleware that manages thing interaction. Approaches for IoT middleware can be found in (Guinard et al., 2011; Kindberg et al., 2002; Mrissa et al., 2015; Spiess et al., 2009) and a programming approach for an IoT system can be found in (Latronico et al., 2015). The IFTTT (https://ifttt.com) approach, is extended to IoT by using a sensor as a trigger, for example, for sending a message. IFTTT is then a way of defining IoT flows. Another programming possibility is the use of Node-RED (https://nodered.org), an event driven, non-blocking model that can be used both at the edge or in the cloud. It enables construction of flows from the wide range of nodes available as a palette.

The Web of Things (https://www.w3.org, W3C WoT-Working Group) has been proposed to improve the interoperability and usability of the Internet of Things (IoT) and has been implemented, for example, as Mozilla WebThings. It gives URLs to connected devices thereby linking them and enabling their discovery. The result is a layer for the IoT that is

Prakash, N. and Prakash, D.
Concepts for Conceptual Modelling of an IoT Application.
DOI: 10.5220/0011082800003176
In *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 494-501
ISBN: 978-989-758-568-5; ISSN: 2184-4895

independent of underlying IoT protocols like Zigbee, DECT ULE, and KNX.

In the second sense (Eterovic et al., 2015), the term, IoT system is used to express an application that uses things. Thus, here an IoT system describes the application for use of both, non-expert users as well as IoT developers. Our interest here is in development activities for building IoT applications and therefore in this second sense.

An IoT application is itself variously described. Huang and Li (Huang et al., 2010) define an IoT application in terms of the "information of thing". An IoT application is one whose "information of thing" is expressed in UID/EPC, is embedded in an RFID electronic tag, and is uploaded by non-machine contact. Clearly, this definition is very low level. Costa (Costa et al., 2016) defines an IoT application as "a collection of automated procedures and data, integrated with heterogeneous entities (hardware, software, and personnel) that interact with each other and with their environment to reach common goals." They also define an 'IoT application system' as a 'composition of Devices and Services interacting with other Devices, Physical Entities, and Users'. Thus, an IoT application system is a prescription for, or a specification of, the IoT to be implemented.

There are two broad approaches to this specification, UML based (Eterovic et al., 2015; Thramboulidis et al., 2016) and SySML based (Costa et al., 2016, Kotronis et al., 2018) respectively. The former follows the software engineering approach and exploits notions of components, ports etc. of component diagrams of UML. The latter adopts the system engineering approach and uses notions of blocks and block interaction through connectors and ports of SySML.

A domain specific UML based language for describing an IoT in terms of things and their interaction was proposed in (Eterovic et al., 2015). In this language, a thing is considered to be the core element of an IoT system. It can be a device, software or a subsystem. A thing contains zero or more items that may be inputs (e.g. temperature), outputs (e.g. light switch} and components (e,g, log service). Things are loosely connected and a change in one of them does not affect the other. Interaction among things is via ports.

In (Costa et al., 2016) device experts convert device properties like types of devices, device features as well as data formats and protocols into components, that is, devices, services and resources of the IoT application system. Further, device experts model the structure and internal configurations of devices in accordance with the IoT Domain Model.

Kotrinis (Kotronis et al., 2018) postulate that there are a number of critical factors, for example, energy consumption and time, that must be taken into account during IoT application system design. The approach adopted is to decompose the application system into sub-systems and associate critical aspects with sub-systems. Devices of subsystems then must satisfy these criticalities. They assume that devices are known upfront (ECG and fall detector in their application) and the BDD and IBD of SysML are used to show interaction.

The commonality in the foregoing is the assumption that devices and their interaction is a given and the problem is to arrive at a suitable representation that forms the implementation design of the IoT. This has a number of drawbacks

- The properties of devices like units, ranges, types of devices etc., are not problem-driven. This may lead to ill-fitting devices to be selected.
- Properties of communication like distance, data quantity and data rates are again decided at implementation time and are not obtained from the problem being addressed.
- There is no explicit link between the device and the real-world entity that it senses/actuates. Thus, even though we know that there is a temperature sensor, there is no explicit expression that the temperature of a boiler is to be measured.
- The nature of the processing to be performed is left unspecified. Thus, if measurements are to be cleaned (Jeffery et al., 2006) before transmission, then this is unspecified. Similarly, system reliability may require multiple measurements of the same property, say pressure. A single value is then obtained from the multiple measurements. However, reliability is an implementation issue rather than being problem-driven.

Our position in this paper is that a problem-driven specification of an IoT application system would overcome these problems. Such a specification performs the same role as done by conceptual modelling in information systems, IS, engineering. By analogy with an IS, we refer to an IoT application system as an Information System of Things, ISoT and a conceptual model as CMoT, Conceptual Model of Things. Following the notion of a conceptual model (Genero et al., 2005), a CMoT provides implementation-independent concepts for a complete expression of the system to be built. Thus, it provides high-level concepts for representing real-world entities, the phenomena to be sensed and its properties, communication and its properties, as well as any processing to be carried out. By analogy with the term, conceptual schema in conceptual modelling, we refer to an instantiation of the CMoT, as a Conceptual Schema of Thing, CSoT. Whereas the implementation design for the former is, for example, the logical schema of a relational data base, the implementation design for a CSoT is for building a collection of devices and their interaction.

The CSoT acts as a shared document among problem/domain experts for discussion and eventually, when accepted, represents an agreement about the system to be developed. Further, it acts as a prescription of the system to be implemented, and thus, is a meeting point between system designers and system implementors. The latter must implement exactly that which is specified in the CSoT.

The notion of a conceptual schema has been used to conceptualize bespoke information systems as well as for integrating schemas of existing information systems. By analogy, we expect the CSoT to also be a conceptualization of a bespoke ISoT and allow CSoT integration.

In this paper, we compare and contrast an ISoT with an IS and propose a set of concepts for a CMoT. The similarities and differences of these concepts with those of IS conceptual modelling are considered. We also illustrate the use of these in conceptualizing a bespoke ISoT with a view to showing the feasibility of our position. We postpone dealing with CSoT integration till after a fully fleshed-in CMoT is developed. Lastly, we do not address the issue of converting a CSoT into implementation design and leave it as future work.

The layout of the paper is as follows. In section 2 we consider the notion of an ISoT and introduce the concepts required for its conceptual modelling. Section 3 deals with modelling issues like aggregation and ISA hierarchies, associations and dependencies. In section 4, we apply our notions to modelling the lighting of a home.

2 CONCEPTUAL MODELLING CONCEPTS

We draw an analogy between an Information System, IS, and an IoT application. As is well known, an IS is embedded in an organization and it is a teleological system that keeps a record of business transactions through a well-defined functionality. For example, an IS for a hotel is for the purpose of room reservation; it keeps a record of reservation transactions (bookings, cancellations); and provides functionality for hotel reservation transactions. It is embedded in the hotel business and receives transaction invocation stimuli from actors in its environment to which it responds. This makes the notion of an object as in object-orientation central to IS conceptual modelling. An IS object sends a message to invoke a method of a receiver object. The receiver responds and the sender proceeds after the response has arrived. The receiver, in turn, awaits further invocation.

Analogous to an IS, for an IoT application system, we define an Information System of Things, ISoT to be a **teleological collection of interacting real world entities embedded in an organization**. It is teleological because it has a defined purpose. This purpose may be

- Providing data for analytics and decision making. For example, data about cycling rates of an air conditioner can be provided for analysis for improved cooling efficiency
- Monitoring and regulating. For example, a temperature sensor may send data to the cooling unit to switch it off/on.

An ISoT is embedded in an environment and the principal interaction with the environment in which it is embedded is to periodically deliver data to it. It does so autonomously, i.e., the data and the rate at which data is sent are determined by the ISoT. There is no "request for data" or "request for function invocation" issued by the environment. Consequently, the question of responding to any request does not arise. This implies that the real-world entities comprising the ISoT are autonomous and (a) may take a decision based on the sensed data, or (b) communicate the sensed data or, perhaps, after processing it. Consider a simple ISoT consisting of a steam boiler. Based on the temperature it senses, the boiler may decide to switch off/on the gas supply and also, send the on/off action taken and the time of the action to its environment.

The absence of a "request" makes the notion of an object ill-fitting for an ISoT. **We propose to use the notion of an agent instead of an object to model real-world entities of an ISoT.** An agent (Wooldridgey et al., 2000) is "situated within or part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future." Applying this, let us define an agent, boiler. This agent is situated in the environment of steam and gas supply and has its own agenda, namely, to regulate the temperature of steam. It senses the temperature of steam and acts to switch on/off the gas supply that in turn, changes the temperature that the boiler senses in the future. The agent boiler operates over time and runs continuously.

Now, we define real world entities as individuals having one or more **aspects** that are sensed/measured. There are two major differences between aspects and attributes as follows:

- Aspects are sensed and, unlike attributes, are not 'describing properties' of entities. Consider a person P. P is of interest in an ISoT for its aspect, blood pressure that is sensed for monitoring. The history of the blood pressure as well as the most recent blood pressure measurement may be required. On the other hand, P is of interest in an IS for its attribute, address that describes it and whose last updated value is required.
- Values of aspects may be false positive or false negative due to sensor errors, or unreliable because sensors may have "failed dirty". Therefore, obtaining the right value requires cleaning (Jeffery et al., 2006; Gonzalez et al., 2007) of the sensed data. However, attribute values do not display this phenomenon.

Aspects and attributes are similar in that

- Both may be constrained. For example, the aspect, blood pressure cannot be more than 280 mm of Hg. Though in both cases a constraint specifies the allowed values, in the case of an ISoT, they may also help in the selection of the appropriate device to be used. Thus, in out example, a blood pressure meter that measures up to 28-mm Hg is required.
- Both attributes and aspects may contain missing data. For the former this may be because of unknown data but for the later, again, this is on account of missed values during sensing (Jeffery et al., 2006).

From the foregoing, we can say that **agents have aspects**. This is consistent with belief-desire-intention approach in agent orientation where, an agent is associated with (Abushark et al., 2017), beliefs or what it knows about itself and its surrounding environment; its desire or what it wants to achieve; and intentions or the ways in which it achieves this. We represent the first of these through the notion of an aspect of an agent. Evidently, an aspect corresponds to the notion of a belief. The desires of an ISoT agent are varied and may be, for example, to clean the sensed data, process the sensed data, communicate with other agents and so on. We model intentions as **functions/procedures** of agents.

The agent, Boiler is illustrated in Fig. 1. It has two aspects, Raw_temp and Raw_pres, for temperature and pressure measurements, respectively. There are two functions, Clean_temp and Clean_pres to smoothen the incoming stream of temperature and pressure measurements respectively. This results in two additional aspects, Proc_temp and Proc_pres for processed temperature and pressure respectively.

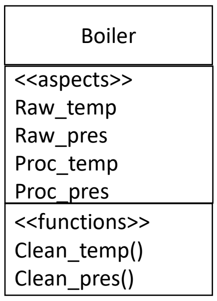

Figure 1: The Agent, Boiler.

As mentioned earlier, we are dealing with **interacting** real world entities. In an IS, this takes the form of functional interaction. On the other hand, in an ISoT, interaction takes the form of communication of messages and who communicates with whom is of interest. From the perspective of agents, it is possible to classify (Shehory et al., 2014; Wooldridgey et al., 2000) agents based on five properties, namely, being communicative, learning ability, mobility (ability to transport itself from one machine to another), flexibility and having a character. Evidently, ISoT interaction requires the property of being communicative. Consequently, we propose using the notion of a COMMunicative AGent or COMMAG.

Now, we model a communication as a 4-tuple, <mode, sender, letter, receiver>. Senders and receivers are COMMAGs; a sender/receiver may be an individual or a group. Additionally, we postulate a special type of COMMAG called a Gateway that receives/sends messages from/to the external world. There are five possible modes as follows: This gives us four additional modes of communication as follows:

1. **Broadcast:** a message to all COMMAGs.
2. **One-to-One**: a boiler sending its temperature to the gas supply unit
3. **One-to-Group**, the boiler sending its message to both, the gas supply unit and the steam receiving unit
4. **Group-to-One**, the several sections of a boiler comprise a group, Boiler, and the Boiler sends the average temperature of the group to the gas supply unit.
5. **Group-to-Group**, the several regions of the boiler send their temperatures to both the gas supply unit and the steam receiving unit

A letter is a specification of the communication between COMMAGs and is in two parts. The first part tells us the properties of the communication. These include the Mode of communication, Distance that contains the distance between the sender and the receiver, and Data Rate. This acts as a selection criterion for the communication standard (Zigbee, Bluetooth etc.) that shall be deployed when the CSoT is implemented. The second, the message part, contains the message itself. This may be data sensed by the sender COMMAG or a processed form of data. Since both parts are describing properties of the communication, we represent these as attributes (NOT aspects) of the communication.

The letter and the communication mode taken together show the number of COMMAGs that shall receive the message. As shown in the examples of the communication modes (1) to (5) above, the CSoT designer needs to provide this information so that appropriate messaging is implemented in the IoT system to be built.

Fig. 2 shows the one-to-one communication between the Boiler and the Gas Supply Unit. The Gas Supply Unit has an aspect, Status to show if it is switched on or off. This is toggled by the function Switch_status. The block arrow shows communication between the Boiler and the Gas Supply Unit. The former sends a letter to the latter and this letter consists of the Mode, Distance and Data Rate, as well as Proc_temp and Proc-pres as the message.

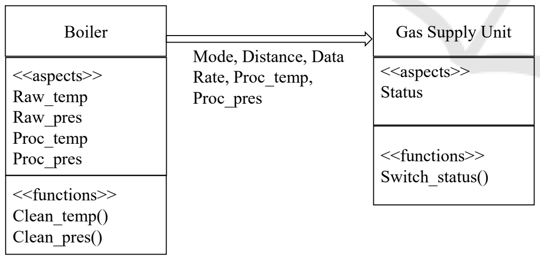

Figure 2: The Communication of the Boiler.

To sum up. we see that an ISoT rests on three principles, autonomy, sensing, and communication. There are five basic concepts for conceptual modelling that takes these into account, namely, **Communicative Agents** that have **Aspects** and **Functions** together with **Constraints** that are defined on aspects. Agents autonomously decide when and what to communicate to other agents. This decision is based on their internal state, as captured in aspects, as well as on their sensory input. Communication between these agents is captured in the notion of **Letters** that contain the message to be sent as well as communication properties.

3 THE MODEL

Let us now consider using the concepts of section 2 to build a model of an ISoT. Again, we do this by analogy with concepts that go into modelling an IS. As is well known, the UML notation is used to represent an IS in three models, namely, the structure model, behaviour or dynamic model, and the function model. In this section we consider the applicability of these to an ISoT.

As already seen, the basic structural unit in an ISoT is the COMMAG. Structurally, a COMMAG of an ISoT may be simple or aggregations just as happens in an IS. For example, the COMMAG, Milk Van is an aggregate of the COMMAGs, Storage section (having aspect, temperature) and a Cooling Unit (with aspect, working status). Similarly, a COMMAG may show generalization/specialization hierarchies. For example, a hypertensive person (aspect, blood pressure) may be specialized into a hypertensive, diabetic person (inherited aspect, blood pressure and specialized aspect, sugar level).

Whereas associations and dependencies are of interest in an IS, this is not the case in an ISoT. First consider associations. An association captures an object-object relationship. This is interesting in an IS because transactions change this relationship and a record of the current relationships is required. For example, the association, Owns between Persons and Houses is of interest in an IS to know ownership. However, associations between COMMAGs are uninteresting because there are no transactions, an association, while giving information on which device is associated with which one, fails to specify either device information or interaction information. For the same reason, a "dependency" is uninteresting from the ISoT perspective.

Now consider behaviour modelling and interaction diagrams. In an IS, functional interaction between objects. as in sequence diagrams, specifies system behaviour. Which object sends a message to which object and the response it receives before proceeding further, gives full details of message passing. In an ISoT, COMMAGs are autonomous. A COMMAG, in fulfilment of its desire to keep other COMMAGs informed, sends out a communication. Having sent out the communication, the COMMAG continues and does not wait for any response. Therefore, there is no need for sequence diagrams. On the other hand, we need a way to express that a message is sent by a COMMAG to another if the former decides to do so. This decision is the trigger for the messaging action (in the IFTTT sense) and such decisions constitute the autonomous capacity of the COMMAG. The arrow between COMMAGs together with the letter suffices to model this.

An IS has a third model for capturing processes through process diagrams. These diagrams carry the logic of the business process of the IS. In an ISoT, process diagrams are useful to show the progression/flow of data through the IoT, from one COMMAG to another. In our boiler example, the reliability of pressure/temperature measurements is crucial for maintaining safety. The designer of the boiler CSoT defines a COMMAG, Boiler_Section that measures pressure and temperature of a section of the boiler. Thus, multiple measurements for multiple boiler sections are made and their average computed before deciding to activate the gas supply unit. This flow of data from sensing through possibly several layers of processing before the data is consumed constitutes the process diagram for an ISoT. Such a diagram is needed for each kind of sensed data (pressure and temperature in our example). When the CSoT is taken into implementation, then an identification of edge/fog and cloud computing would be possible.

The remaining question is as to how communication is represented. We propose to show this as a directed edge from the sender to the receiver with communication attributes attached to it. This implies that there is only one diagram that shows:

- A COMMAG with its aspects and functions
- The aggregate and specialization/generalization hierarchies of COMMAGs
- The communication edge between COMMAGs and its attributes.

Simple, aggregate entities together with generalization/specialization specify the nature of devices that must be associated with real-world entities.

4 AN EXAMPLE

In this section, we illustrate conceptual modelling of an ISoT through a small example to illustrate the viability of our position.

The aim is to control the lighting, of a house. The house consists of a bedroom and a large living room that is divided into sections, each having its own lighting arrangement. There are several residents who interact with the house. The lighting requirement is as follows:

1 Light in the bedroom and living room is based on the ambient luminosity and set to the value desired by the residents.
2 Bedroom light is time controlled. It is switched off at a preset clock time.
3 Each section of the living room is occupancy controlled. If occupied, then it is lighted else the light is switched off.

The conceptual schema of the house is shown in Fig. 3. ISA hierarchy and aggregation are shown using UML notation. As shown in the figure, we define a COMMAG, Room that has an aspect, Light_intensity and a function, Light control to set Light_intensity to the desired value. The desired value is communicated by the COMMAG, Resident. Room is specialized into two COMMAGs, namely, (a) Bedroom and (b) Living Room. The sections of the living room are represented by the COMMAG, LRsection. The living room is an aggregate of its sections as shown by the aggregate COMMAG, Livingroom. It has a function to determine the average luminosity of the room from the luminosity of its several sections.

LRsection has an aspect, occupancy. This is used by the function Occ-L-Control to control light of the section based on occupancy.

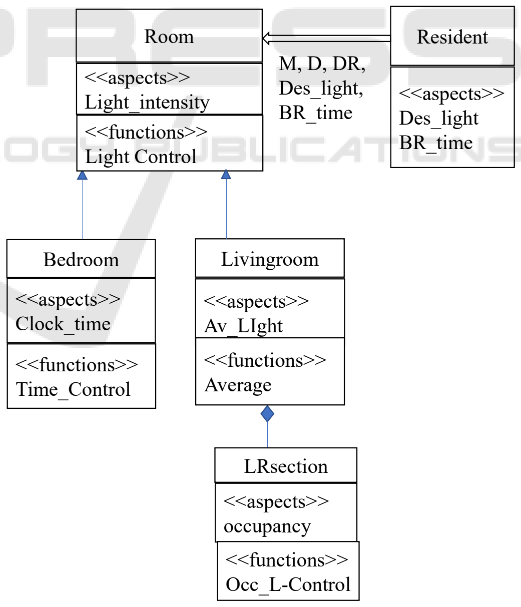

Figure 3: Lighting Control.

The COMMAG bedroom has an as aspect clock time. It is communicated by Resident as bedroom time, BR_time. There is a corresponding function to control the bedroom light based on this. Lastly, Resident has two aspects, one for the desired light value, Des_light and the other for bedroom clock time, BR_time as discussed above.

The letter for communication between Resident and Room consists of communication properties as follows

(a) Mode of communication, M that is the broadcast mode
(b) Distance, D of communication. For house bound residents this distance may be 10 metres whereas for office-going residents it may be in the kilometre range, say 15 kilometres.
(c) Data Rate, DR. Since the message consists of about 8 bytes of information, and clearly, data rates are very low and set to 8Bps.

Further, message itself consists of Des_light and BR_time.

This example shows that as in traditional conceptual modelling, specification of constraints must be done by annotation. Notice that there is no message passing to invoke a method. The letter sent, besides specifying communication attributes sends data that are not function arguments. The receiver agent can decide what action is to be taken, if any. Thus, LRsection decides autonomously to switch off/on a light or to continue leaving it off or on, despite having Des-light communicated to it by the Resident.

5 CONCLUSIONS

Our position is that in order to make device and communication decisions that fit well with the desired IoT application, we should develop a conceptual schema of the application. The conceptual schema is a specification of the implementation to be carried out in the subsequent stage. By analogy with Information Systems, we propose an ISoT as a collection of communicating agents that is embedded in an environment. The ISoT communicates with the environment through the agent, Gateway.

The ISoT is substantially different from an IS because of its emphasis on data exchange as different from function invocation in an IS. It behaves autonomously based on what is senses and is therefore, naturally amenable to being modelled in agent-orientation. The Belief-Desire-Intention triad of agent orientation can be extended to an ISoT.

The set of concepts for IoT conceptual modelling are again somewhat different from those of IS conceptual modelling. Our communicating agents have aspects that are measurable properties and not describing properties that IS objects have. There is no need for associations and dependencies but only for aggregates and ISA hierarchies. Further, sequence diagrams are not interesting since there is no method invocation. Process diagrams do not carry business logic but show the transformation of data as it moves from measurement to final consumption. However, attributes are needed for specifying communication.

When looked at from the point of view of an IoT system as interacting things, the CSoT defines the needed devices, processing and communication to be built. Devices are identified from aspects of COMMAGs, and the measurement ranges of these devices is identified from constraints on aspects. Processing is identified from functions of COMMAGs and, when building thing interaction, a decision needs to be taken about its location, whether at the edge, fog or cloud. Finally, a letter consists of the properties of communication that help in decisions about the underlying communication standards/protocols to be used. The consequence is that the implemented system is better aligned to the application at hand.

In future we expect to fully flesh out the CMoT, and look into converting it into an implementation design. Further, we expect to look into aspects of CSoT integration and use it as a basis for agile development of an ISoT.

REFERENCES

Abushark, Y., Thangarajah, J., Harland, J., & Miller, T. (2017). A framework for automatically ensuring the conformance of agent designs. Journal of Systems and Software, 131, 266-310.

Costa, B., Pires, P. F., & Delicato, F. C. (2016, August). Modeling IoT applications with sysml4iot. In 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 157-164). IEEE.

Eterovic, T., Kaljic, E., Donko, D., Salihbegovic, A., & Ribic, S. (2015, October). An Internet of Things visual domain specific modeling language based on UML. In 2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT) (pp. 1-5). IEEE.

Genero, M., Piattini, M., & Calero, C. (Eds.). (2005). Metrics for software conceptual models. World Scientific.

Gonzalez, H., Han, J., & Shen, X. (2007, April). Cost-conscious cleaning of massive RFID data sets. In 2007 IEEE 23rd International Conference on Data Engineering (pp. 1268-1272). IEEE.

Guinard, D., Trifa, V., Mattern, F., & Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In

Architecting the Internet of things (pp. 97-129). Springer, Berlin, Heidelberg.

Hachem, S., Teixeira, T., & Issarny, V. (2011, December). Ontologies for the internet of things. In Proceedings of the 8th middleware doctoral symposium (pp. 1-6).

Huang, Y., & Li, G. (2010, August). Descriptive models for Internet of Things. In 2010 International Conference on Intelligent Control and Information Processing (pp. 483-486). IEEE.

Jeffery, S. R., Alonso, G., Franklin, M. J., Hong, W., & Widom, J. (2006, May). Declarative support for sensor data cleaning. In International Conference on Pervasive Computing (pp. 83-100). Springer, Berlin, Heidelberg.

Keoh, S. L., Kumar, S. S., & Tschofenig, H. (2014). Securing the internet of things: A standardization perspective. IEEE Internet of things Journal, 1(3), 265-275.

Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., ... & Spasojevic, M. (2002). People, places, things: Web presence for the real world. Mobile Networks and Applications, 7(5), 365-376.

Kotronis, C., Nikolaidou, M., Dimitrakopoulos, G., Anagnostopoulos, D., Amira, A., & Bensaali, F. (2018, June). A model-based approach for managing criticality requirements in e-health iot systems. In 2018 13th annual conference on system of systems engineering (SoSE) (pp. 60-67). IEEE.

Latronico, E., Lee, E. A., Lohstroh, M., Shaver, C., Wasicek, A., & Weber, M. (2015). A vision of swarmlets. IEEE Internet Computing, 19(2), 20-28.

Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015, December). Internet of things (IoT) security: Current status, challenges and prospective measures. In 2015 10th international conference for internet technology and secured transactions (ICITST) (pp. 336-341). IEEE.

Mrissa, M., Médini, L., Jamont, J. P., Le Sommer, N., & Laplace, J. (2015). An avatar architecture for the web of things. IEEE Internet Computing, 19(2), 30-38.

Shehory, O., & Sturm, A. (2014). A brief introduction to agents. In Agent-Oriented Software Engineering (pp. 3-11). Springer, Berlin, Heidelberg.

Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., De Souza, L. M. S., & Trifa, V. (2009, July). SOA-based integration of the internet of things in enterprise services. In 2009 IEEE international conference on web services (pp. 968-975). IEEE.

Thramboulidis, K., & Christoulakis, F. (2016). UML4IoT—A UML-based approach to exploit IoT in cyber-physical manufacturing systems. Computers in Industry, 82, 259-272.

Wang, W., De, S., Toenjes, R., Reetz, E., & Moessner, K. (2012, June). A comprehensive ontology for knowledge representation in the internet of things. In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 1793-1798). IEEE.

Wooldridgey, M., & Ciancarini, P. (2000, June). Agent-oriented software engineering: The state of the art. In International workshop on agent-oriented software engineering (pp. 1-28). Springer, Berlin, Heidelberg..

Yanwei, S., Guangzhou, Z., & Haitao, P. (2011, December). Research on the context model of intelligent interaction system in the internet of things. In 2011 IEEE International Symposium on IT in Medicine and Education (Vol. 2, pp. 379-382). IEEE.

Zhong, C. L., Zhu, Z., & Huang, R. G. (2015, August). Study on the IOT architecture and gateway technology. In 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES) (pp. 196-199). IEEE.