Knowledge Based Interaction of Smart Metering Service Capabilities

Ivaylo Ivanov Atanasov*

Abstract

Service capabilities are reusable software modules that provide specific functions and may be shared by different applications. The paper presents an approach to resolving interactions between service capabilities in the area of smart metering. By using abstraction, semantic information for home energy saving and prepaid smart metering is identified. The information is structured as resources, described by their states and operations that trigger the state changes. Description logic is used to formalize the model of semantic information. Service capabilities are described by logical statements. The service capability interaction is resolved by policies, where policy rules refine the service capability behaviour.

Keywords: Machine-to-Machine Communications, Smart Metering, Knowledge Management, Description Logic

Introduction

Recent service development platforms exploit the idea of reusable software modules that encapsulate generic functions. The idea was born with the introduction of the Intelligent Network concept, where service independent building blocks represent generic functions in the telecommunication network. Next, it was developed in open service access which is standardised as service architecture for mobile networks, and where service capability features are network functions accessible by an application via open interfaces. The same idea was adopted in the area of machine-to-machine (M2M) communications which allow interconnection between smart objects (devices with sensors and/ or actuators) and network infrastructure (ETSI TR 102 725 v.1.1, 2013). M2M network applications share functions exposed by service capabilities through a set of open interfaces and they are software modules that provide specific services, such as location and status of a particular device (Hernandez & Reiff-Marganiec, 2014).

The variety of applications that may be developed by the use of software modules representing specific functions introduces the problem with service interaction. A lot of research has been done in the area of resolving the problem with undesired behaviour caused by interaction of two or more services (Seigmund *et al*., 2012; Chentouf & Khoumsi, 2013; Apel *et al*., 2013; Takeyama & Chiba, 2013). A lot of services use policies to express operational criteria and policies might contradict each other (Maternaghan & Turner, 2013). Mechanisms for discovering and resolving service interaction and policy conflicts are of main importance in service deployment. Description logic appears to be a suitable tool for description of policies and policy-based service logics (Schutte & Wahl, 2010; Hu *et al*., 2012; Tang *et al*., 2014). Description logic is a mathematical formalism used for description of concepts and relations (roles) in semantic networks (Zhang *et al*., 2013; Krotzsch *et al*., 2014). In this paper, an approach is presented to resolve interaction between M2M service capabilities whose logic is based on policies. Service capability interactions are resolved by reasoning about consistency of logical statements.

---

* Faculty of Telecommunications, Technical University of Sofia, Bulgaria. Email: iia@tu-sofia.bg

In the M2M functional architecture, service capabilities are exposed to M2M applications through REST web services (ETSI TS 102 690). REST stands for REpresentational State Transfer and it is architectural style in programming where client-server applications exchange information about resources (Asad *et al.*, 2013; Laine, 2011; Wang *et al.*, 2011). Each resource is accessed by uniform resource identifier and has a state which can be manipulated by four interactions: create, retrieve, update and delete. HTTP protocol is used in many cases. This paper considers service capabilities in the M2M application area of smart metering.

Smart meters are electronic measurement devices used by utilities to communicate information for billing customers and operating their electric systems. The combination of the electronic meters with two-way communications technology for information, monitor, and control is commonly referred to as Advanced Metering Infrastructure (EEI and AEIC, 2011). Smart meters have prepaid functionality which enables optimisation of customer energy consumption and efficient monitoring and management distribution networks (Geisler, 2013; Hwang & Yoon, 2014; Ramlee *et al.*, 2013; Wu & Yu, 2013). Smart metering information systems may be used for remote control of home appliances used for heating and cooling (Erol-Kantarci & Mouftah, 2013; Lim *et al.*, 2013; Liu, 2013; Reaidy *et al.*, 2014). In this paper, an abstraction of semantic data is modelled and captured in resources meaningful for prepaid smart metering and residential cooling/heating control. The resource and their relationships are described as concepts and roles and service capabilities which manipulate the resource states are described by a set of logical statements.

The paper is structured as follows. First, a model of prepaid customer balance is proposed which considers the smart meter prepaid functionality. Next, a model of controllable air conditioner is suggested which applies heating and cooling control. The description of both models is formalised by using description logic. A good introduction to description logics may be found in the paper by Baader (2010). Service capabilities interaction which is based on policy is illustrated for heating and cooling control of air conditioners owned by prepaid customers.

Modeling of Resources in Prepaid Smart Metering

Identification of Requirements to the Support of Prepaid Smart Metering

Prepayment functionality requires real-time control of energy usage (ETSI TR 102 691, 2010). It shall be possible to apply energy usage monitoring for the accumulated usage of energy by a consumer on per predefined daily periods. This capability is required for enforcing dynamic policy decisions based on the total energy usage in real time. The smart metering information system that uses energy usage monitoring for making dynamic policy decisions shall set and send the applicable thresholds to the smart meters for monitoring. The energy usage monitoring thresholds shall be based either on time or on volume. The smart metering information system may send both thresholds to the smart meters. The smart meter shall notify the smart metering information system when a threshold is reached and report the accumulated usage since the last report for energy usage monitoring. If both time and volume thresholds were provided to the smart meter, the accumulated energy usage since last report shall be reported when either the time or the volume thresholds are reached.

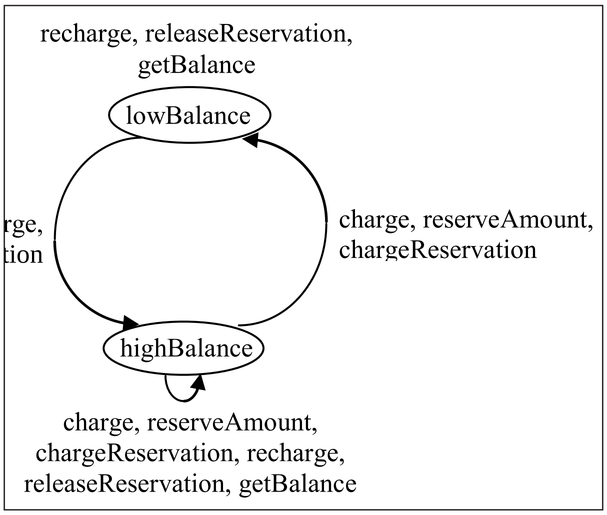
Usage monitoring on smart meter level is active provided that certain conditions are met. For energy usage monitoring at smart meter level, the applicable condition is provisioning of a volume and/or time threshold and activation of at least one rule critical peak periods. Energy usage monitoring allows to connect a supply and to disconnect when volume and/or time threshold is reached.

Management of customer account requires access to account information for update and query operations. Balance update may be result of recharging the account by the customer. Account management includes setting up notifications for account changes and delivering of notifications about account balance changes. M2M applications involved in energy management may reserve appropriate amount in the customer account to ensure that the customer can fulfill his payment obligations. The prepaid account is periodically charged against the reservation. The reservation may be enlarged. In case of unused reserved amount, the reservation shall be released and the remaining amount shall be available for future use.

Prepaid Customer Account as a Resource

Looking at the prepaid customer account as a resource, two resource states are meaningful – an account with low balance, which is near to the monitoring thresholds, and account with high balance which is sufficient for prepaid energy usage. Figure 1 shows the model of prepaid customer account.

Figure 1: A State Model of Prepaid Customer Account



The customer account is in *low Balance* state when it is near to the balance threshold. When the customer recharges the account he updates the account balance. The balance type identifies an existing balance type in the account or a new balance type to be added to the account. Account recharging results in entering into *high Balance* state. In *high Balance* state, energy consumption based on volume and/or time leads to changes in the customer account. Smart metering information system may directly charge the customer account or make a reservation and next charge the reservation. Unused reserved amount can be returned to the account. The reservation release returns funds left in a reservation.

In the domain of prepaid functionality, the prepaid customer account states become concepts and the operations that trigger transitions from one state to another are roles. The T Box consists of the statements representing the transitions in the state model of the prepaid customer account. The statements (1)-(3) define transitions when the customer recharges his prepaid account.

$$\text{lowBalance} \sqsubseteq \exists \text{recharge.highBalance} \tag{1}$$

which says that, if the balance in the customer account is low, the account may be recharged and the balance becomes high.

$$\text{lowBalance} \sqsubseteq \exists \text{recharge.lowBalance} \tag{2}$$

which says that, if the balance in the customer account is low, the account may be recharged and the balance may remain low.

$$\text{highBalance} \sqsubseteq \exists \text{recharge.highBalance} \tag{3}$$

which says that, if the balance in the customer account is high, the account may be recharged and the balance remains high.

In any state, a query about current account balance may be issued, which is defined by statements (4)-(5).

$$\text{lowBalance} \sqsubseteq \exists \text{getBalance.lowBalance} \tag{4}$$

$$\text{highBalance} \sqsubseteq \exists \text{getBalance.highBalance} \tag{5}$$

Charging the customer's account results in reducing the current balance. Statements (6)-(7) define the corresponding transitions.

$$\text{highBalance} \sqsubseteq \exists \text{charge.lowBalance} \tag{6}$$

which says that, if the balance in the customer account is high, the account may be charged and the balance may become low.

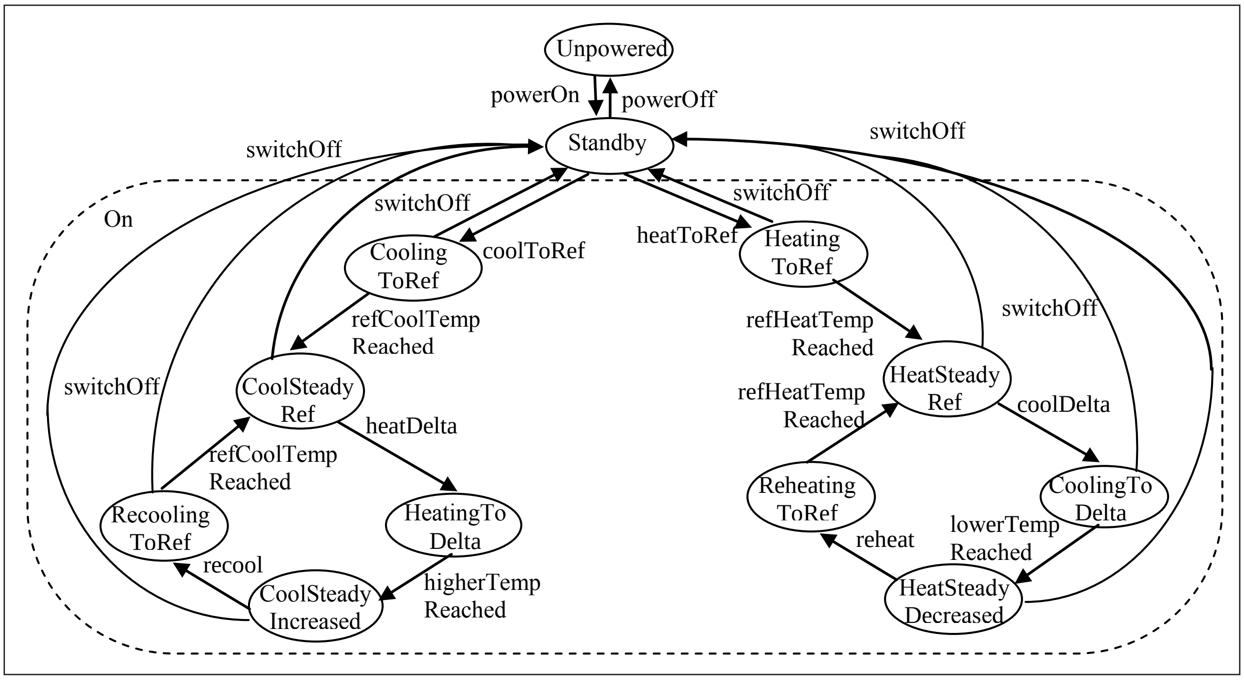
$$\text{highBalance} \sqsubseteq \exists \text{charge.highBalance} \tag{7}$$

which says that, if the balance in the customer account is high, the account may be charged and the balance may remain high.

The amount reservation is preferred when the charging entity is unable to determine beforehand whether the service could be delivered or when the requested number of resources is not known prior to the use of service. The reservation is charged when the reserved resources are exhausted. If there are unused resources within the reservation, the reservation is in order to refund the remaining amount. Statements (8)-(13) define the transitions related to reservation mode of charging.

$$\text{highBalance} \sqsubseteq \exists \text{reserveAmount.highBalance} \tag{8}$$

Figure 2: **State Model of an Air Conditioner**



which says that, if the balance in the customer account is high, the amount reservation may result in high balance.

$$\text{highBalance} \sqsubseteq \exists \text{reserveAmount.lowBalance} \tag{9}$$

which says that, if the balance in the customer account is high, the amount reservation may result in low balance.

$$\text{highBalance} \sqsubseteq \exists \text{chargeReservation.highBalance} \tag{10}$$

which says that, if the balance in the customer account is high, the reservation charging may result in high balance.

$$\text{highBalance} \sqsubseteq \exists \text{chargeReservation.lowBalance} \tag{11}$$

which says that, if the balance in the customer account is high, the reservation charging may result in low balance.

$$\text{highBalance} \sqsubseteq \exists \text{releaseReservation.highBalance} \tag{12}$$

which says that, if the balance in the customer account is high, the account balance remains high in case of reservation release.

$$\text{lowBalance} \sqsubseteq \exists \text{releaseReservation.highBalance} \tag{13}$$

which says that, if the balance in the customer account is low, the reservation release may cause balance increase.

Modeling of Energy Management

Identification of Requirements to Heating/ Cooling Control

Home energy saving applications are aimed at reducing energy consumption at home. Only electrical appliances that can work in autonomous mode and have three states (off, on and standby) may be controlled remotely. In the domain of home energy saving, the concepts represent the air conditioner state and the roles indicate the control operations.

The heating/ cooling control reflects the tolerance in the thermal comfort of the home owner. It allows reducing in case of heating or increasing in case of cooling the preferred (referenced) temperature T_{ref} with a small amount δ of several degrees during the day periods when there is nobody at home or everybody sleeps. Te heating/ cooling control results in energy saving as it is discussed in the following paragraph where the remote control on an air conditioner is explained by the air conditioner states.

Figure 2 shows the state model of an air conditioner which is under the heating/ cooling control. The following explanation is for the cooling control (left part in Figure 2). The same considerations are applicable to the heating control (right part in Figure 2).

The air conditioner is considered to be an intelligent device capable to communication with network applications through communication network. When the air conditioner is powered on, it is in the *Standby* state or in *On* state. When the air conditioner is in *Standby* state it is ready to receive instructions. When the air conditioner is in the *On* state can be controlled remotely. The *On* state is a composite state (denoted by the dash symbol in Figure 2). The transition from *Standby* state to *On* state is triggered by the switchOn operation which is equivalent to cool To Ref in case of cooling or heat To Ref in case of heating.

Considering cooling functionality, a network application instructs the air conditioner to cool to the referenced temperature Tref and the air conditioner's cooling system is switched on. The air conditioner state becomes to *CoolingToRef* state while cooling and it consumes power P_c in order to reach T_{ref}. The air conditioner sends a notification to the network application when the reference temperature T_{ref} is reached and move to *CoolSteadyRef* state. While the air conditioner maintains the referenced temperature it's the cooling system consumes power P_{ref}, where $P_{ref} < P_c$. Following energy saving schedule defined by the home owner, the network application sends instructions to heat by small amount δ during the daily period when nobody is at home. On receiving heatToDelta command, the cooling system of the air conditioner is switched off and now power is consumed. The air conditioner state is *HeatingToDelta* while the heating to $T_{ref\text{-}\delta}$. The air conditioner sends a notification to the network application when the temperature is increased by δ and its state becomes *CoolSteadyIncreased*. The air conditioner consumes power P_{low} where $P_{low} < P_{ref}$ while it maintains the higher temperature $T_{ref\text{-}\delta}$. The network application sends to the air conditioner instructions to recool to the reference temperature T_{ref} at the end of the energy saving period. The air conditioner state becomes *RecoolingToRef*. When the preferred temperature T_{ref} is reached the air conditioner sends a notification to the network application and its state becomes *CoolSteadyRef* again. This procedure is repeated for each energy-saving period.

Controllable Electrical Appliance as a Resource

The concepts and roles related to heating/ cooling control are shown in Table 1.

Table 1: Concepts and Roles Related to Air Conditioner Control

Unpowered	*The air conditioner is powered off and can not be controlled remotely.*
powerOff	Operation that is used to power off the air conditioner.
powerOn	Operation that is used to power on the air conditioner.
Standby	The air conditioner is in a standby state ready to receive instructions.
switchOff	Operation that instructs the air conditioner to switch off it's heating/cooling system.
goToRef	Operation that instructs the air conditioner to heat/cool to the reference temperature.
GoingToRef	Represents the air conditioner state while heating/cooling to the reference temperature.
refTempReached	Notification that the reference temperature is reached.
SteadyRef	Represents the air conditioner state in which the reference temperature is maintained.
goToSave	Operation that instructs the air conditioner to switch off its heating/cooling system in order to maintain the lower/higher temperature that saves energy.
GoingToSaved	Represents the air conditioner state in which its heating/cooling system is switched off in order to reach the lower/higher temperature that saves energy.
saveTempReached	Notification that the lower/higher temperature that saves energy is reached.
SteadySaved	Represents the air conditioner state in which the lower/higher temperature is maintained.
returnToRef	Operation that instructs the air conditioner to reheat/recool to the referenced temperature.
ReturningToRef	Represents the air conditioner state in which its heating/cooling system is switched on in order to reach the reference temperature.

The logical statements that describe the relationships between concepts and roles and represent the transitions in the air conditioner state model are shown by equations (14)-(26):

$$\text{Unpowered} \sqsubseteq \exists \text{powerOn.Standby.} \tag{14}$$

$$\text{Standby} \sqsubseteq \exists \text{powerOff.Unpowered.} \tag{15}$$

$$\text{Standby} \sqsubseteq \exists \text{goToRef.GoingToRef.} \tag{16}$$

$$\text{GoingToRef} \sqsubseteq \exists \text{refTempReached.SteadyRef.} \tag{17}$$

$$\text{SteadyRef} \sqsubseteq \exists \text{goToSave.GoingToSaved.} \tag{18}$$

$$\text{GoingToSaved} \sqsubseteq \exists \text{saveTempReached.SteadySaved.} \tag{19}$$

$$\text{SteadySaved} \sqsubseteq \exists \text{returnToRef.ReturningToRef.} \tag{20}$$

$$\text{ReturningToRef} \sqsubseteq \exists \text{refTempReached.SteadyRef.} \tag{21}$$

$$\text{GoingToRef} \sqsubseteq \exists \text{switchOff.Standby.} \tag{22}$$

$$\text{SteadyRef} \sqsubseteq \exists \text{switchOff.Standby.} \tag{23}$$

$$\text{GoingToSave} \sqsubseteq \exists \text{switchOff.Standby.} \tag{24}$$

$$\text{SteadySaved} \sqsubseteq \exists \text{switchOff.Standby.} \tag{25}$$

$$\text{ReturningToRef} \sqsubseteq \exists \text{switchOff.Standby.} \tag{26}$$

The ABox contains the statement (27) saying that initially all air conditioners are unpowered:

$$s_0: \sqcap_{h\backslash CSET}\,(\text{Unpowered}), \tag{27}$$

where s_0 is the air conditioner initial state and CSET is the set of all air conditioners.

The air conditioner may be in exactly one state at any moment which is expressed by (28):

$$\top \sqsubseteq_{\backslash} (\sqcup_{s1,s2\backslash CSTATES,\ s1\backslash s2}(s1 \sqcap s2)) \sqcap (\sqcup_{s2\backslash CSTATES}\, s) \tag{28}$$

where CSTATES denotes the air conditioner states.

The equations (29)-(35) describe the allowed functions in each state by action functions

$$\text{Func(Unpowered)}=\{\text{powerOn}\}. \tag{29}$$

$$\text{Func(Standby)}= \{\text{heatToRef}\}\backslash\{\text{powerOff}\} \tag{30}$$

$$\text{Func(GoingToRef)}=\{\text{refTempReached}\}\backslash\{\text{switchOff}\} \tag{31}$$

$$\text{Func(SteadyRef)}= \{\text{goToSave}\}\backslash\{\text{switchOff}\} \tag{32}$$

$$\text{Func(GoingToSave)} = \{\text{saveTempReached}\}\backslash\{\text{switchOff}\} \tag{33}$$

$$\text{Func(SteadySaved)} =\{\text{returnToRef}\}\backslash\{\text{switchOff}\} \tag{34}$$

$$\text{Func(ReturningToRef)} = \{\text{ref Temp Reached}\}\backslash\{\text{switch Off}\} \tag{35}$$

The fact that the air conditioner state is changed the only by certain actions is represented (36):

$$\text{For all } s\backslash\text{CSTATES, and all } R\backslash\text{Func}(s),\ s \sqsubseteq \backslash R.s \tag{36}$$

Resolving Interaction Between Service Capabilities

Grounds for Using Descriptive Logic as a Tool for Service Logic Description

Service capability interaction occurs when the integration of two service capabilities would modify the behaviour of one or both service capabilities. Service capability interaction concerns certain behaviour of interacting capabilities such as how their response time may be changed given the integration. For example, let us consider interaction between heating/cooling control service capability and prepaid account management service capability. Undesired behaviour may occur when the heating/ cooling control service capability requests reheating or re-cooling to the reference temperature which requires energy consumption, and the prepaid account management service capability provides information that the current account balance is low.

There are three fronts on which service capability interaction problem can be attacked: avoidance, detection, and resolution. In some cases, it is possible to design service control model in such a way that certain interactions cannot occur. Service capability interaction problem may be avoided at the phase of new service development by checking its interaction with other known services. When a conflict between service capabilities occurs it has to be resolved. Resolving service capability interaction problem may be based on policies (Kumar *et al.*, 2012; Pencheva, 2013). The policies define rules that determine the service capability behaviour in case of conflicts. Descriptive logic is a suitable tool for definition of policy rules which describe the behaviour in case of service capability interaction.

Policy based Interaction between Service Capabilities

Let us consider the Prepaid Energy Saving Service (PESS) which applies heating/cooling control for prepaid customers. The PESS integrates functionality of heating/cooling control service capability and prepaid account management service capability. In addition, it is assumed that the PESS preliminary can calculate the amount to be charged. The policy rules for PESS related to activation of heating/cooling control for prepaid customers are defined by replacing the equation (16) with equations (37) to (37).

$$\text{Standby} \sqcap \text{highBalance} \sqsubseteq \exists \text{goToRef}.\text{GoingToRef} \sqcap \exists \text{charge}.\text{highBalance}, \quad (37)$$

which says that, if PESS is active and the air conditioner is in *Standby* state and the account balance is high, then the air conditioner may be instructed to reach the reference temperature and it moves to *GoingToRef* state and the after charging the balance remains high.

$$\text{Standby} \sqcap \text{lowBalance} \sqsubseteq \exists \text{goToRef}.\text{Standby} \quad (38)$$

which says that, if PEES is active and the air conditioner is in *Standby* state and the account balance is low, then the air conditioner may be instructed to reach the reference temperature and it remains in *Standby* state.

$$\text{Standby} \sqcap \text{highBalance} \sqsubseteq \exists \text{goToRef}.\text{GoingToRef} \sqcap \exists \text{charge}.\text{lowBalance} \quad (39)$$

which says that, if PESS is active and the air conditioner is in *Standby* state and the account balance is high, then the air conditioner may be instructed to reach the reference temperature and it moves to *GoingToRef* state and after charging the balance becomes low.

Following the same approach the equation (20) which requires returning to the reference temperature also is replaced with equations (40) to (42).

$$\text{SteadySaved} \sqcap \text{highBalance} \sqsubseteq \exists \text{return To Ref}.\text{Returning To Ref} \sqcap \exists \text{ charge. high Balance}, \quad (40)$$

which says that, if the air conditioner is in *SteadySaved* state and the balance is high, the air conditioner may be instructed to re-heat/re-cool to the reference temperature and it moves to *ReturningToRef* state and after charging the balance remains high.

$$\text{SteadySaved} \sqcap \text{lowBalance} \sqsubseteq \exists \text{returnToRef}.\text{SteadySaved}, \quad (41)$$

which says that, if the air conditioner is in *SteadySaved* state and the balance is low, the air conditioner may be instructed to re-heat/re-cool to the reference temperature and it remains in *SteadySaved* state.

$$\text{SteadySaved} \sqcap \text{highBalance} \sqsubseteq \exists \text{returnToRef}.\text{ReturningToRef} \sqcap \exists \text{charge}.\text{lowBalance}, \quad (42)$$

which says that, if the air conditioner is in *SteadySaved* state and the balance is high, the air conditioner may be instructed to re-heat/ re-cool to the reference temperature and it moves to *ReturningToRef* state and after charging the balance becomes low.

Other example of service capability interaction represents interaction between heating/cooling control and control of windows' and door's states. Heating/ cooling control over an air conditioner may be applied only when windows and the door of the room are closed.

Conclusion

The paper presents an approach to resolving service capability interaction based on policies. By using abstraction in the area of smart metering, semantic information about resources related to heating/ cooling control and prepaid smart metering functionality is defined. The resources are described by their states and operations that manipulate the resource states. The resource description is formalised by means of description logic where resource state model is presented by a set of logical statements. The service capability interaction can be resolved by policy-based logics. Policy rules refine the description of the behaviour of service capabilities in case of interaction.

The method for modeling information about smart metering resources is in conformance of REST architectural style which is suitable for constrained devices.

References

Apel, S., Rhein, A., Thum, T., & Kastner, C. (2013). Feature-interaction detection based on feature-based specifications. *The International Journal of Computer and telecommunications Networking*, 57(12), 2399-2409.

Asad, O., Erol-Kan Kumar, M., & Mouftah, H. T. (2013). A survey of sensor web services for the smart grid. *MDPI Journal of Sensor and Actuator Networks,* 2(1), 98-108.

Baader, F. (2010). *The Description Logic Handbook*. Cambridge University Press.

Chentouf, Z., & Khoumsi, A. (2013). A high abstraction level approach to detecting feature interactions between telecommunication services. *Journal of King Saud University – Computer and Information Sciences,* 25(1), 99-115.

EEI & AEIC (2011). A Discussion of Smart meters And RF Exposure Issues. An EEI-AEIC-UTC White paper. A Joint project of the EEI and AEIC meter Committees. Retrieved from http://aeic.org/wp-content/uploads/2013/07/smartmetersandrf031511.pdf

Erol-Kantarci, M., & Mouftah, H. T. (2013). *Smart Grid Forensic Science: Applications, Challenges and Open Issues*. IEEE Communications Magazine Feature Topic on Ultimate Technologies and Advances for Future Smart Grid UTASG, 51(1), 68-74.

ETSI TR 102 691 (2010). Machine-to-Machine communications (M2M). Smart Metering Use Cases, v1.1.1.

ETSI TR 102 725 (2013). Machine-to-Machine communications (M2M). Definitions, v1.1.1.

ETSI TS 102 690 (2011). Machine-to-Machine communications (M2M). Functional architecture, v1.1.1.

Geisler, K. (2013). The Relationship between Smart Grids and Smart Cities. IEEE Smart Grid. Retrieved from http://smartgrid.ieee.org/may-2013/869-the-relationship-between-smart-grids-and-smart-cities

Hernandez, M., & Reiff-Marganiec, S. (2014). Classifying smart objects using capabilities, *SmartComp*. Retrieved from http://www.cs.le.ac.uk/people/srm13/publications/smartcomp14.pdf.

Hu, L., Qiu, C., & Shi, Y. (2012). A defeasible description logic based semantic security policy conflict detection approach. *International Journal of Security and Its Applications*, 6(2), 313-319.

Hwang, K., & Yoon, S. H. (2014). A multifunctional RF remote control for ultralow standby power home appliances. *International Journal of Distributed Sensor Network*. Retrieved from http://dx.doi.org/10.1155/2014/381430

Krotzsch, M., Simancik, F., & Horrocks, I. (2014). Description Logics. *IEEE Intelligent Systems*, 29(1), 12-19.

Kumar, V., Fensel, A., Lazendic, G., & Lehner, U. (2012). Semantic policy-based data management for energy efficient smart buildings. *Lecture Notes in Computer Science*, 7567, 272-281.

Laine, M. (2011). RESTful web services for the internet of things, (pp.1-3). Retrieved from http://media.tkk.fi/webservices/personnel/markku_laine/restful_web_services_for_the_internet_of_things.pdf

Lim, M., Bahr, W., & Leung, S. (2013). RFID in the warehouse: A literature analysis (1995-2010) of its applications, benefits, challenges and future trends. *International Journal of Production Economics*, 145(3), 409-430.

Liu, Y. (2013). Study on smart home system based on internet of things technology. *Informatics and Management Science* IV, W. Du, Ed., (207, pp. 73-81). Springer London.

Maternaghan, C., & Turner, K. (2013). Policy conflicts in home automation. *Computer Networks*, 57(12), 2429-2441.

Pencheva, E. (2013). Policy based web services interaction management. *International Journal of Computer Science and Mobile Computing*, 2(12), 55-63.

Ramlee, R. A., Leong, M., Singh, R., Ismail, M., Othman, M., & Sulaiman, H. (2013). Bluetooth remote home automation system using android application. *International Journal of Engineering and Science,* 2(1), 149-153.

Reaidy, P., Gunasekaran, A., & Spalanzani, A. (2014). Bottom-up approach based on internet of things for order fulfilment in a collaborative warehousing environment. *International Journal of Production Economics*. DOI: 10.1016/j.ijpe.2014.02.017.

Schutte, J., & Wahl, T. (2009). Inter-domain Policy Conflicts. *IEEE Vehicular Technology Magazine,* 5(3), 68-74.

Seigmund, N., Kolesnikov, S., Kastner, C., Apel, S., Batory, D., Rosenmuller, M., & Saake, G. (2012). *Predicting Performance via Automated Feature-Interaction Detection*. Proceedings of IEEE International Conference on Software Engineering, pp.167-177.

Takeyama, F., & Chiba, S. (2013). Implementing feature interactions with generic feature modules. *Software Composition. Lecture Notes in Computer Science,* 8088, 81-96.

Tang, C., Wang, L., Tang, S., Qiang, B., & Tian, J. (2014). Semantic description and verification of security policy based on ontology. *Wuhan University Journal of Natural Sciences*, 19(5), 385-392.

Wang, M., Fan, C., Wen, Z., Liu, S., & Li, J. (2011). *Implementation of Internet of Things Oriented Data Sharing Platform Based on RESTful Web Service*. Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing WiCOM (pp.1-4).

Wu, G., & Yu, X. (2013). *Remote Control System for Energy Efficient Home. Proceedings of IEEE Conference Energytech*,(pp.1-5).

Zhang, J., Wang, W., Shi, Z., Tian, D., Yue, J., & Zhang, B. (2013). *Dynamic Description Logic Based Semantics Web Service Composition*. Proceedings of International Conference on Semantics, Knowledge and Grids, (pp.194-197).