Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 12, December 2014, pg.376 – 383

RESEARCH ARTICLE

Knowledge Building for Home Energy Saving Domain

Ivaylo Atanasov

Faculty of Telecommunications, Technical University of Sofia, Bulgaria

iia@tu-sofia.bg

Abstract— The paper proposes an approach to building semantic knowledge for home energy saving, which appears to be main Internet of Things application. Description logic is used to formalize knowledge base representation. Service modelling is considered as a kind of knowledge management which allows reasoning and resolving service interaction.

Keywords— Internet of Things, Home automation, Semantic Information, Knowledge management, Description Logic

---

I. INTRODUCTION

Internet of Things (IoT) is the next evolutionary step in the development of Internet technology. IoT covers communications between smart objects with Internet Protocol connectivity, which leverage between the digital and physical world. Smart objects are small microelectronic devices that contain communication module, a microprocessor and a sensor or actuator. Recently, the traffic of such devices is constantly increasing in many different areas such as medicine, transportation, energy efficiency and more. Network architecture for smart objects should be open to future innovation. Innovation must be possible for both the usage of smart objects and the way of designing the IoT technology. The whole architecture is the foundation and it must be extremely flexible while maintaining new applications in the future, just as the Internet itself did for the last four decades [1], [2], [3], [4].

IoT communications sets new requirements for network architecture, based on autonomous communication between sensors/actuators and the corresponding application on the Internet [5], [6]. An autonomous network is formed by the federation of heterogeneous systems, self-formed without human intervention and specific infrastructures. Autonomic network is characterized by self-management and forms its own policies and decisions in accordance with the information received from the environment in which it operates.

Development of models and methods for self-management can be aided by the synthesis of semantic information. Ability of IoT system to enable applications to discover, interpret and use IoT data is essential for the creation of IoT services of high level and for the development of an open market for IoT data [7], [8], [9], [10], [11]. To enable transmission of IoT data of a system, it is necessary to synthesize semantic information. Through such semantic information, applications can discover data without prior knowledge of them. Applying cognitive techniques, the semantic information may be used for planning, learning, and acting according to end to end goals.

This paper presents an approach to building semantic knowledge for home energy saving, which appears to be one of the main IoT applications. Recent research on semantic information about home energy saving applications studies high level architectural principles for modelling semantic information or concerns specific implementation [12], [13], [14], [15], [16], [17]. As far as the author's knowledge there is a lack of structural approaches to derive knowledge from semantic information which may be used for the development of autonomous agents embedding function for self-configuring and self-optimization, which cooperate in the context of the managed system. In addition, the suggested approach to modeling semantic information may be used for resolving service interactions by the use of policies. Description logic appears to be a suitable tool to define policy rules that refine service behavior in case of interactions [18].

The paper is structure as follows. First, the knowledge about electrical appliances, premises and home energy saving is formalized. Next, services for home energy saving are modeled by refinement. Modeling services by descriptive logic may be useful in resolving service interaction. The conclusion summarizes the contribution and the benefits of the proposed solution.

II. MODELLING OF SEMANTIC INFORMATION FOR HOME ENERGY SAVING

Descriptive logic is a formal language that may be used for knowledge description and reasoning. The Knowledge base consists of a Terminology box (TBox) and Assertion box (ABox). The Tbox introduces terminology i.e. the vocabulary in the application domain, while the ABox contains statements related to facts defined in the vocabulary. The vocabulary consists of concepts, which represent set of entities and roles, which denote binary relationships between entities.

In the domain of home energy saving, it is important to abstract semantic information about controllable electrical appliances used for cooling and heating, as well as for the premises where the appliances are located.

The proposed model of a home energy saving system is composed of an Appliance-category subsystem and of a Room-category subsystem. In the context of energy saving, only the electrical appliances which convert energy (e.g. energy is converted in lighting, cooling or heating) are of interest. The Appliance category subsystem is characterized by Appliance usage, Usage mode and Appliance state. The appliance usage can be periodic (e.g. washing machine usage), permanent (e.g. air conditioner, hot water boiler and ventilation system usage) or semi-random (e.g. personal computer usage). The appliance usage mode can be autonomous (e.g. air conditioner, hot water boiler and ventilation system usage mode) or manual/user directed (TV set top box, personal computer). Only the controllable appliances which have autonomous usage mode and permanent appliance usage may be subject of power control for the aim of energy saving. The appliance state may be unpowered and powered, where the power on and power off are manual operations. The powered state of controllable appliance is composite state which may be standby state or on state. In standby state the appliance is powered on and it does not make energy conversion. In this state the appliance is powered on and it makes energy conversion. Examples of controllable electrical appliances are air conditioners, hot water tanks and ventilation systems.

Each type of electrical appliances and its states are represented as concepts. The roles represent the operation on controllable electrical appliances (e.g. switchOn and switchOff operations used for transitions between the standby state and on state).

The Tbox consists of the statements representing the hierarchy of electrical appliances. For example, statements describing the appliance state are given by (1) to (8).

$$\text{ElectricalAppliance} \sqsubseteq \exists \text{Has.ApplianceState}, \tag{1}$$
which says, that electrical appliance has state.

$$\text{ApplianceState} \sqsubseteq \text{Unpowered} \sqcup \text{Powered}, \tag{2}$$
which says, that appliance state is Powered or Unpowered.

$$\text{Unpowered} \equiv \neg \text{Powered}, \tag{3}$$
which says, that both appliance states are mutual exclusive.

$$\text{Unpowered} \sqsubseteq \exists \text{powerOn.Powered}, \tag{4}$$
which says, that the appliance moves from Unpowered state to Powered state by applying powerOn operation.

$$\text{Powered} \sqsubseteq \exists \text{powerOff.Unpowered}, \tag{5}$$
which says, that the appliance moves from Powered state to Unpowered state by applying powerOff operation.

$$\text{Powered} \sqsubseteq \text{On} \sqcup \text{Standby}, \tag{6}$$
which says, that Powered state is On or Standby.

$$\text{Standby} \sqsubseteq \exists \text{switchOn.On}, \tag{7}$$
which says, that the appliance move from Standby state to On state by applying switchOn operation.

$$\text{On} \sqsubseteq \exists \text{switchOff.Standby}, \tag{8}$$
which says, that the appliance move from On state to Standby state by applying switchOff operation.

Each electrical appliance is located in a room. The following concepts are used to represent room characteristics relevant to home energy saving. Container is an object that contains anything. Room is a kind of container. Window is a part of a room. Door is a part of a room. DoorState is the state of a door. WindowState is a state of a window. WOpen represents an open window. WClosed represents a closed window. DOpen represents an open door, and DClosed represents a closed door.

Fig. 1 illustrates a part of semantic network related to room with controllable appliances.

The Tbox contains statements representing the relationships between room concepts (e.g. statements from (9) to (16)).

$\text{Container} \sqsubseteq \exists \text{contains.ElectricalAppliance}$, (9)
which says that, the container contains an electrical appliance.

$\text{Room} \sqsubseteq \text{Container}$, (10)
which says that, the room is a container,

$\text{Room} \sqsubseteq \exists \text{Has.Door}$, (11)
which says that, the room has a door.

$\text{Door} \sqsubseteq \exists \text{Has.DoorState}$, (12)
which says that, the door has a door's state.

$\text{DoorState} \sqsubseteq \text{DOpen} \sqcup \text{DClosed}$, (13)
which says that, the door's state is open or closed
$\text{DClosed} \equiv \neg \text{DOpen}$. (14)

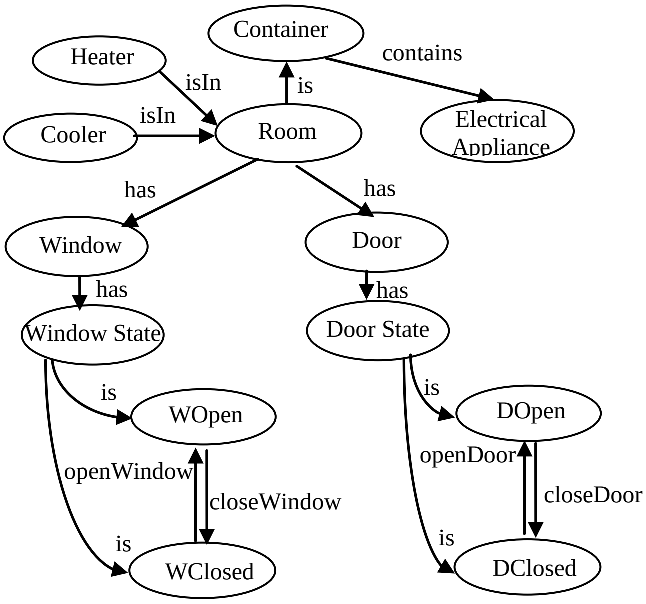

Fig.1. Semantic network related to rooms

$\text{DOpen} \sqsubseteq \exists \text{closeDoor.DClosed}$, (15)
which says that, in DOpen state, when the door is closed and it moves to DClosed state.

$\text{DClosed} \sqsubseteq \exists \text{openDoor.DOpen}$, (16)
which says that, in DClosed state, when the door is opened and it moves to DOpen state.

Let us denote by DSET the set of all doors in a room. By DSTATES we denote the room door states s^d_i. The assertion box contains the statements declaring that initially all doors are closed (s^d_0 denotes the initial state of each door).

$s^d_0 : \sqcap_{d \in \text{DSET}} (\text{DClosed})$. (17)

To express the fact that each door is in exactly one state at any moment we use the statements:

$\top \sqsubseteq \neg(\sqcup s^d_{1,} s^d_{2 \in} \text{DSTATES},\ s^d_{1 \neq s2} (s^d_1 \sqcap s^d_2)) \sqcap (\sqcup s^d_{2 \in} \text{DSTATES}\ s^d)$. (18)

The door state changes by means of actions defined as action functions. An action function for given state corresponds to the possible transitions in the DSTATE. The action functions related to door state are Func(DClosed) = {openDoor}, and Func (DOpen) = {closeDoor}.

The fact that each door can change the state only by means of certain actions is represented by the following statement:

For all $s^d \in \text{DSTATES}$, and all $R \notin \text{Func}(s^d)$, $s^d \sqsubseteq \forall R.s^d$

III. MODELLING KNOWLEDGE FOR POWER CONTROL

The power control aimed at energy saving reflects the maximum tolerance assumptions of home owner for changes in home minimal / maximal temperature (i.e. Δ). For a simple family house, the day temperature changes Δ may vary as follows: -1^{o}C (in winter) or $+1^{o}$C (in summer) during breakfast, lunch and dinner time, -2^{o}C (in winter) or $+2^{o}$C (in summer) during periods when children are at school or when everybody is sleeping).

Fig.2 shows the state model of a cooler which is under the cooling control.

Let us assume that in the beginning the heater is powered on but its heating system is switched off (i.e. the heater is in the Standby state ready to receive instructions) and the temperature in the room where the heater is located is 30^{0}C. When the cooling system is switched on, it receives instructions to heat the room to the reference temperature T_{ref} (20^{0}C) and it results in transition to CoolingToRef state. The heating system consumes power P_h in order to reach T_{ref}. When the reference temperature T_{ref} is reached (the transition to CoolSteadyRef state), the cooling system consumes power P_{ref} maintaining the reference temperature, where $P_{ref} < P_h$. Following preliminary defined schedule, power control is applied. Control command to heat by small amount of time Δ (e.g. 2^{0}C) results in switching off the cooling system in order to cool to higher temperature $T_{ref-\Delta}$. While cooling to $T_{ref-\Delta}$ no power is consumed (CoolingToDelta state). When the temperature reaches $T_{ref-\Delta}$, the cooling system is switched on in order to maintain the higher temperature $T_{ref-\Delta}$. During the energy saving period the cooling system consumes lower power P_{low} (CoolSteadyIncreased state). At the end of the energy saving period, the cooler receives instructions to re-cool to the reference temperature which results in transition to RecoolingToRef state. When the referenced temperature is reached (CoolSteadyRef state), the cooling system consumes power P_h again and the cycle is closed.

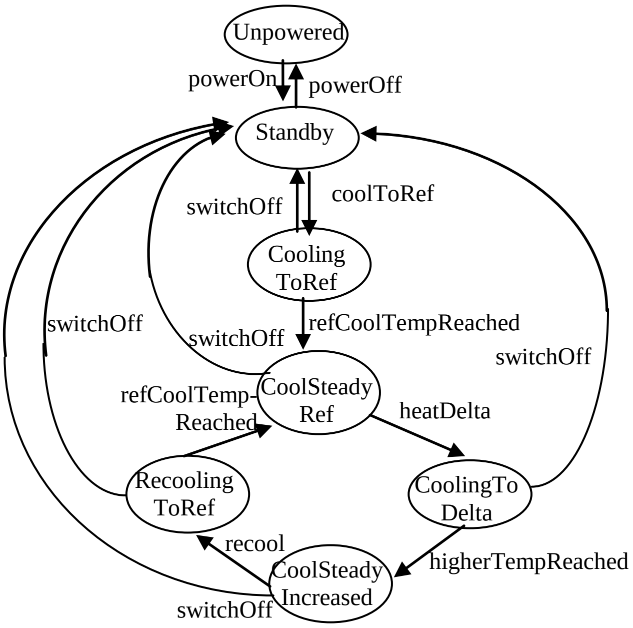

Fig.2. State model of a cooler

Additional concepts and roles related to cooling control are defined. The Tbox for cooling control includes the following statements representing the transitions in the cooler state model.

$\text{On} \sqsubseteq \text{CoolingToRef} \sqcup \text{CoolSeadyRef} \sqcup \text{HeatingToDelta} \sqcup \text{CoolSteadyIncreased} \sqcup \text{RecoolToRef}.$ (19)

which says that, when the cooler is in On stated, it is in CoolingToRef state, or CoolSeadyRef state, or HeatingToDelta state, or CoolSteadyIncreased state, or RecoolToRef state.

$\text{coolToRef} \equiv \text{switchOn},$ (20)

which says that, switchOn operation is equivalent to coolToRef operation (for coolers).

$\text{Standby} \sqsubseteq \exists \text{coolToRef.CoolingToRef},$ (21)

which says that, if the cooler is in Standby state, it may be instructed to cool to the reference temperature and it moves to CoolingToRef state.

$\text{CoolingToRef} \sqsubseteq \exists \text{refCoolTempReached.CoolSeadyRef},$ (22)

which says that, if the cooler is in CoolingToRef state and the reference temperature is reached, it moves to CoolSeadyRef state.

$\text{CoolSeadyRef} \sqsubseteq \exists \text{heatDelta.HeatingToDelta},$ (23)

which says that, if the cooler is in CoolSeadyRef state, it may be instructed to heat to the higher temperature and it moves to HeatingToDelta state.

$\mathrm{HeatingingToDelta} \sqsubseteq \exists \mathrm{higherTempReached.CoolSteadyIncreased},$ (24)
which says that, if the cooler is in HeatingToDelta state and the higher temperature is reached, it moves to CoolSteadyIncreased state.

$\mathrm{CoolSteadyIncreased} \sqsubseteq \exists \mathrm{recool.RecoolingToRef},$ (25)
which says that, if the cooler is in CoolSteadyIncreased state, it may be instructed to re-cool to the reference temperature and it moves to RecoolingToRef state.

$\mathrm{RecoolingToRef} \sqsubseteq \exists \mathrm{refCoolTempReached.CoolSeadyRef},$ (26)
which says that, if the cooler is in RecoolingToRef state and the reference temperature is reached, it moves to CoolSteadyRef state.

The statements from (27) to (31) say that in any On state, the cooler may be switched off and it moves to Standby state.

$\mathrm{CoolingToRef} \sqsubseteq \exists \mathrm{switchOff.Standby}.$ (27)

$\mathrm{CoolSeadyRef} \sqsubseteq \exists \mathrm{switchOff.Standby}.$ (28)

$\mathrm{HeatingToDelta} \sqsubseteq \exists \mathrm{switchOff.Standby}.$ (29)

$\mathrm{CoolSteadyIncresed} \sqsubseteq \exists \mathrm{switchOff.Standby}.$ (30)

$\mathrm{ReCoolingToRef} \sqsubseteq \exists \mathrm{switchOff.Standby}.$ (31)

Let us denote by CSET the set of all coolers and by CSTATES the states s_i of a cooler. The assertion box contains the following statement, where s_0 denote the initial state for each cooler.

$s_0 \colon \sqcap_{h \in \mathrm{CSET}} (\mathrm{Unpowered}),$ (32)

which says that initially all coolers are unpowered.
To express the fact that each cooler is in exactly one state at any moment we use the statement:

$\top \sqsubseteq \neg(\sqcup_{s1,s2 \in \mathrm{CSTATES},\ s1 \neq s2}(s1 \sqcap s2)) \sqcap (\sqcup_{s2 \in \mathrm{CSTATES}}\ s)$ (33)

The cooler's state changes by means of actions defined as action functions. An action function for given state corresponds to the possible transitions in the CSTATE. The following action functions for cooling control are defined:

$\mathrm{Func(Unpowered)} = \{\mathrm{powerOn}\}.$ (34)
$\mathrm{Func\ (Standby)} = \{\mathrm{powerOff}\} \cup \{\mathrm{heatToRef}\}.$ (35)
$\mathrm{Func\ (CoolingToRef)} = \{\mathrm{switchOff}\} \cup \{\mathrm{refCoolTempReached}\}.$ (36)
$\mathrm{Func\ (CoolSteadyRef)} = \{\mathrm{switchOff}\} \cup \{\mathrm{heatDelta}\}.$ (37)
$\mathrm{Func\ (HeatingToDelta)} = \{\mathrm{switchOff}\} \cup \{\mathrm{higherTempReached}\}.$ (38)
$\mathrm{Func\ (CoolSteadyIncreased)} = \{\mathrm{switchOff}\} \cup \{\mathrm{recool}\}.$ (39)
$\mathrm{Func\ (RecoolingToRef)} = \{\mathrm{switchOff}\} \cup \{\mathrm{refCoolTempReached}\}.$ (40)

The fact that each cooler can change the state only by means of certain actions is represented by the following statement:

For all $s \in \mathrm{CSTATES}$, and all $R \notin \mathrm{Func}(s)$, $s \sqsubseteq \forall R.s$

IV. SERVICE MODELLING AS A KIND OF KNOWLEDGE MANAGEMENT

Services are modeled by *refinement*. The definition of refinement is formalized as refinement operation δ_F, for given service *App*, which operation transforms given knowledge base K into another knowledge base $\delta_{App}(K)$. The last is augmented by a set of *activation concepts* which generally are $A_{App} \subseteq \{ App_h \mid h \in \mathrm{CSET}\}$. We use contexts $C[\varphi]$ to define refinements in the knowledge base, where φ is a subformula of any formula ψ.

The Cooling Control Service (CCS) applies cooling control in a room with a cooler. If a door or a window of the room is open, no cooling control is applied. If cooling control is applied, and the window or door of the room is open, the cooler is switched off. The refinement statements for CCS are defined by equations (41) to (62).

$C_1[\neg \mathrm{CCS}_H \sqcap \mathrm{Standby}_H \sqcap (\mathrm{WOpen}_H \sqcup \mathrm{DOpen}_H)] \sqsubseteq \exists \mathrm{coolToRef}.C_2[\mathrm{CoolingToRef}_H],$ (41)

which says that, if CCS is not active and the cooler is in Standby state and the window or the door is open, then when the cooler is instructed to cool to the referenced temperature, it moves to CoolingToRef state.

$$C_1[CCS_H \sqcap Standby_H \sqcap (WOpen_H \sqcup DOpen_H] \sqsubseteq \exists coolToRef.\ C_2[Standby_H], \tag{42}$$

which says that, if CCS is active and the cooler is in Standby state and the window or the door is open, then when the cooler is instructed to cool to the referenced temperature, it moves to Standby state.

$$C_3[\neg CCS_H \sqcap CoolingToRef_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_4[DOpen_H] \tag{43}$$

which says that, if CCS is not active and the cooler is in CoolingToRef state and door is closed, then the door may be opened.

$$C_3[\neg CCS_H \sqcap CoolingToRef_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_4[WOpen_H], \tag{44}$$

which says that, if CCS is not active and the cooler is in CoolingToRef state and the window is closed, then the window may be opened.

$$C_3[CCS_H \sqcap CoolingToRef_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_4[DOpen_H] \sqcap \exists switchOff.\ {}_4[Standby_H], \tag{45}$$

which says that, if CCS is active and the cooler is in CoolingToRef state and the door is closed, then the door may be opened and the cooler is switched off.

$$C_3[CCS_H \sqcap CoolingToRef_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_4[WOpen_H] \sqcap \exists switchOff.\ C_4[Standby_H], \tag{46}$$

which says that, if CCS is active and the cooler is in CoolingToRef state and the window is closed, then the window may be opened and the cooler is switched off.

$$C_5[\neg CCS_H \sqcap CoolSeadyRef_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_6[DOpen_H], \tag{47}$$

which says that, if CCS is not active and the cooler is in CoolSeadyRef state and door is closed, then the door may be opened.

$$C_5[\neg CCS_H \sqcap CoolSeadyRef_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_6[WOpen_H], \tag{48}$$

which says that, if CCS is not active and the cooler is in CoolSeadyRef state and the window is closed, then the window may be opened and the cooler is switched off.

$$C_5[CCS_H \sqcap CoolSeadyRef_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_6[DOpen_H] \sqcap \exists switchOff.C_6[Standby_H], \tag{49}$$

which says that, if CCS is active and the cooler is in HeatSeadyRef state and the door is closed, then the door may be opened and the cooler is switched off.

$$C_5[CCS_H \sqcap HeatSeadyRef_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_6[WOpen_H] \sqcap \exists switchOff.C_6[Standby_H], \tag{50}$$

which says that, if CCS is not active and the cooler is in CoolSeadyRef state and the window is closed, then the window may be opened and the cooler is switched off.

$$C_7[\neg CCS_H \sqcap HeatingToDelta_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_8[DOpen_H], \tag{51}$$

which says that, if CCS is not active and the cooler is in HeatingToDelta state and the window and door are closed, then the door may be opened.

$$C_7[\neg CCS_H \sqcap HeatingToDelta_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_8[WOpen_H], \tag{52}$$

which says that, if CCS is active and the cooler is in HeatingToDelta state and the window and door are closed, then the window may be opened.

$$C_7[CCS_H \sqcap HeatingToDelta_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_8[DOpen_H] \sqcap \exists switchOff.C_8[Standby_H], \tag{53}$$

which says that, if CCS is active and the cooler is in HeatingToDelta state and the door is closed, then the door may be opened and the cooler is switched off.

$$C_7[CCS_H \sqcap HeatingToDelta_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_8[WOpen_H] \sqcap \exists switchOff.C_8[Standby_H], \tag{54}$$

which says that, if CCS is not active and the cooler is in HeatingToDelta state and the window and door are closed, then the window may be opened and the cooler is switched off.

$$C_9[\neg CCS_H \sqcap CoolSteadyIncreased_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_{10}[DOpen_H], \tag{55}$$

which says that, if CCS is not active and the cooler is in CoolSteadyIncreased state and the door is closed, then the door may be opened.

$$C_9[\neg CCS_H \sqcap CoolSteadyIncreased_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_{10}[WOpen_H], \tag{56}$$

which says that, if CCS is not active and the cooler is in CoolSteadyIncreased state and the window is closed, then the window may be opened.

$$C_9[CCS_H \sqcap CoolSteadyIncreased_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_{10}[DOpen_H] \sqcap \exists switchOff.C_{10}[Standby_H]_, \tag{57}$$

which says that, if CCS is active and the cooler is in CoolSteadyIncreased state and the door is closed, then the door may be opened and the cooler is switched off.

$$C_9[CCS_H \sqcap CoolSteadyIncreased_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_{10}[WOpen_H] \sqcap \exists switchOff.C_{10}[Standby_H]_, \tag{58}$$

which says that, if CCS is active and the cooler is in CoolSteadyIncreased state and the window is closed, then the window may be opened and the cooler is switched off.

$$C_{11}[\neg CCS_H \sqcap RecoolingToRef_H \sqcap DClosed] \sqsubseteq \exists openDoor.C_{12}[DOpen_H], \tag{59}$$

which says that, if CCS is not active and the cooler is in RecoolingToRef state and door is closed, then the door may be opened.

$$C_{11}[\neg CCS_H \sqcap RecoolingToRef_H \sqcap WClosed_H \sqcap DClosed_H] \sqsubseteq \exists openWindow.C_{12}[WOpen_H], \tag{60}$$

which says that, if CCS is not active and the cooler is in RecoolingToRef state and the window is closed, then the window may be opened.

$$C_{11}[CCS_H \sqcap RecoolingToRef_H \sqcap DClosed_H] \sqsubseteq \exists openDoor.C_{12}[DOpen_H] \sqcap \exists switchOff.C_{12}[Standby_H]_, \tag{61}$$

which says that, if CCS is active and the cooler is in RecoolingToRef state and the door is closed, then the door may be opened and the cooler is switched off.

$$C_{11}[CCS_H \sqcap RecoolingToRef_H \sqcap WClosed_H] \sqsubseteq \exists openWindow.C_{12}[WOpen_H] \sqcap \exists switchOff.C_{12}[Standby_H]_, \tag{62}$$

which says that, if CCS is active and the cooler is in RecoolingToRef state and the window is closed, then the window may be opened and the cooler is switched off.

More complicated service logic may consider the cooling control behavior in case of closing the room’s door or window. For this purpose, the cooler state model, shown in Figure 2, must be extended with a Suspended state. In any On state, the cooler may be suspended. In Suspended state, the cooling system of the cooler is switched off. The application implementing the service logic remembers the actual On state in which the cooler is suspended, and may request a transition from Suspended state to the previous On state.

V. CONCLUSIONS

The ubiquitous penetration of IoT in our every day life causes new challenges related to gathering row data from a huge number of heterogeneous devices, deriving information and using the knowledge for particular aims. These call for new methods for data aggregation, information processing and knowledge management. In this paper we propose a structural approach to modelling semantic information for home energy saving and to knowledge management for modelling IoT services. The relevant level of abstraction for IoT data is not limited to particular sensors and actuators. It abstracts the presentation of the physical entity being sensed and acted upon by devices. The entity presentation is generic, inherent to the environment and is not bind to a specific IoT application. Semantic information provides to IoT applications knowledge about electrical appliances details such as meaningful appliance characteristics, actions performed by appliances and interpretation of data delivered by appliances, as well as the premises where the appliances are contained in. We use the logical-based formalism to describe the knowledge in the area of power control for the aim of home energy saving. Innovative services for power control are defined by knowledge refinement and may be used for formal reasoning on the concepts of the application domain of power control.

The suggested approach is applicable to discovering service interaction which manifests itself as undesired behaviour in parallel execution of two on more services. With services described by description logic, the service interaction problem may be considered as a satisfiability problem.

A knowledge base solution gives the opportunity of ubiquitous access and thus increases flexibility. It increases the productivity at the stage of service implementation and maintenance allowing cost reductions, and provides scalability which means service growth upon demand.

REFERENCES

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

[2] K. Geisler, “The Relationship between Smart Grids and Smart Cities”, *IEEE Smart Grid*, 2013, from http://smartgrid.ieee.org/may-2013/869-the-relationship-between-smart-grids-and-smart-cities

[3] O. Asad, M.Erol-Kan Kumar, H.T. Mouftah. “A Survey of Sensor Web Services for the Smart Grid”, MDPI *Journal of Sensor and Actuator Networks*, vol. 2, issue 1, pp. 98- 108, 2013.

[4] M.Erol-Kantarci, H.T. Mouftah, “Smart Grid Forensic Science: Applications, Challenges and Open Issues”, *IEEE Communications Magazine Feature Topic on Ultimate Technologies and Advances for Future Smart Grid UTASG*, vol. 51, number 1, pp. 68-74, 2013.

[5] V. Foteinos, D.Kelaidonis, G. Poulios, P. Vlacheas, V. Stavroulaki, P. Demestichas. “Cognitive Management for the Internet of Things: A Framework for Enabling Autonomous Applications”, *IEEE Vehicular Technology Magazine*, vol.8, issue 4, pp.90-99, 2013.

[6] D. Kyriazis, T. Varvarigou. “Smart, Autonomous and Reliable Internet of Things”, *Procedia Computer Science*, vol 21, pp. 442–448, 2013.

[7] K. O.Aduda, W. Zeiler, G. Boxem, T. Labeodan, “On Defining Information and Communication Technology Requirements and Associated Challenges for ‘Energy and Comfort Active’ Buildings”, *Procedia Computer Science*, vol.32, pp.979-984, 2014.

[8] V.Kumar, A.Fensel, G.Lazendic, U.Lehner, “Semantic Policy-Based Data Management for Energy Efficient Smart Buildings”. OTM Workshop, *Lecture Notes in Computer Science*, vol.7567, pp.272-281, 2012.

[9] Z.Qunzhi, S.Natarajan, Y.Simmhan, V. Prasanna, “Semantic Information Modeling for Emerging Applications in Smart Grid”, *Proc. of International Conference on Information Technology: New Generations ITNG’2012,* pp. 775-782, 2012.

[10] S.Rohjan, M. Uslar, M. Specht, A.Niesse, M. Troschel. “Towards Semantic Service Integration for Automation in Smart grids”, *International Journal of Distributed Energy Sources*, vol.8 number 2, pp.119-146, 2012.

[11] Q.Zhou, Y. Simmhan, V. Prasanna. “Incorporating Semantic Knowledge into Dynamic Data Processing for Smart Power Grids”, *Lecture Notes in Computer Science*, vol.7650, pp.257-273, 2012.

[12] C. H.Hung, Y. W. Bai, P. W.Chen, J.-M. Hsu, “Remote control of home lighting devices with RFID identification and current detection of outlets”, *Proc. of the IEEE 17th International Symposium on Consumer Electronics ISCE '13*, pp.7-8, 2013.

[13] K.-il Hwang, S.-H. Yoon. “A Multifunctional RF Remote Control for Ultralow Standby Power Home Appliances”, *International Journal of Distributed Sensor Networks*, vol.2014, http://dx.doi.org/10.1155/2014/381430

[14] Y. Liu. “Study on Smart Home System Based on Internet of Things Technology”. *Informatics and Management Science* IV, vol. 207, Springer London, pp. 73-81, 2013.

[15] R.A. Ramlee, M. Leong, R.Singh, M.Ismail, M. Othman, H. Sulaiman, et al. “Bluetooth Remote Home Automation System Using Android Application”, *International Journal of Engineering and Science*, vol. 2, pp.149-153, 2013.

[16] P. Reaidy, A.Gunasekaran, A. Spalanzani. “Bottom-up approach based on Internet of Things for order fulfilment in a collaborative warehousing environment” *International Journal of Production Economics*, 2014

[17] G.Wu, X.Yu, “Remote control system for energy efficient home”, *Proc. of IEEE Conference Energytech*, pp.1-5, 2013.

[18] E. Pencheva. “Policy Based Web Services Interaction Management”, International Journal of Computer Science and Mobile Computing, vol.2, issue 12, pp.55-63, 2013.