Mobile Edge Computing Applications for Connectivity Management

Evelina Pencheva, Ivaylo Atanasov
Technical University of Sofia
Sofia, Bulgaria
{enp,iia}@tu-sofia.bg

Abstract—**Connectivity management is the ability to connect and manage mobile devices in Machine-to-Machine (M2M) communications. Resilient and scalable connectivity management, which is fundamental to M2M solutions, may be achieved by using Mobile Edge Computing (MEC) technology. MEC enables applications to timely response to dynamic changes in radio conditions and thus to improve effectiveness of connectivity management. In this paper, we present models of device connectivity management that may be supported by MEC applications. We suggest a method for automatic detection of undesired interaction between applications using standard reasoning over description logic.**

I. Introduction

Mobile Edge Computing (MEC) is a technology that brings the IT service environment and cloud computing capabilities into the Radio Access Network [1]. The close proximity to mobile devices reduces latency and creates a better quality of experience for end users [2], [3]. MEC accelerates applications with real-time requirements which may improve effectiveness of radio resource usage. Connectivity management applications for mobile devices are good candidates for deployment onto a MEC platform.

Connectivity management is the ability to connect and manage mobile devices in Machine-to-Machine (M2M) communications. Resilient and scalable connectivity management is fundamental to M2M solutions. Connectivity management of mobile M2M devices is a complex task when dealing with various network protocols, physical or virtual interfaces. The reasons include the scale of devices (the huge number of devices to be connected), device variations (availability of different categories, models and vendors), criticality of the services (e.g. healthcare application or industrial control), regulation compliance, and performance issues.

Different protocols and proprietary solutions have fragmented the M2M market and have added complexity, time and cost to integration process [4], [5]. An abstraction, required for scalable platform that adheres to standards and addresses a broad range of common M2M functions, is provided by OMA Lightweight M2M [6]. On the other hand, the explosion of M2M services and applications may result in feature interaction. Feature interaction manifests itself as a function of services which is neither exactly the sum of every service nor behaves as expected [7]. Instances of the feature interaction problem have been studied in different M2M applications like home automation [8], automotive systems [9], service systems [10] and in other fields. The compositionality and modularity [11] are in the base of the problem instances, while the difference between the individual views, interpretations and eventual solutions, is considerable. An example for such significant difference might be given when comparing the views on feature interactions of automotive systems engineering and of service systems in aspects like functionality, parallelism, structure, etc.

Despite of the progress in developing approaches for modeling, detecting, and resolving feature interactions, there is a lack of sufficient knowledge on the kind of feature interactions that occur in real-world M2M systems [12]. In our previous works, we studied different aspects of feature interaction in CAMEL networks [13]-[16]. Customized Applications for Mobile Enhanced Logic (CAMEL) is service delivery platform for GSM and UMTS networks. Our research focused on human call related behavior. In [13] and [14], we studied feature interactions based on CAMEL originating and terminating basic call models respectively and reasoned on interactions between services available for calling and called party. In [15] and [16], we stressed on CAMEL mobility management models to study interaction between services as a result of subscriber mobility. CAMEL models are not applicable in the world of M2M communications where devices are used for data transfer.

In this paper, we present models of M2M device connectivity management which are extended with application logic for bearer selection based on different policies. Models are formally described and verified. Further, using semantic abstraction of connectivity management, we use description logic to model policy-based applications for connectivity management. Feature interactions are considered as a contradiction problem and may be discovered automatically by standard reasoning algorithm on description logic.

The paper is structured as follows. In Section II, we briefly present the OMA Trap Framework which allows interoperable way for device management using any kind of events worthwhile for managing and monitoring the networked services or applications deployed on devices, or faults on the general software and hardware, etc. Section III presents device connectivity management models, which are formally described and verified. In Section IV, semantic annotation of

ISSN 2305-7254

device connectivity management is used to describe different applications which add value to bearer selection procedure. The algorithm for inference of feature interaction is presented in Section V. The conclusion summarizes the contribution.

II. OMA DIAGNOSTIC AND MONITORING TRAPS

The OMA Lightweight M2M protocol (LWM2M) is targeted at constrained devices with embedded low power microcontrollers and limited amount of memory, as well as at more powerful embedded devices. It sets a protocol between a server located in a public or private data center, and a client which resides on the device. The interface between the LWM2M client and server allows efficient device management. The focus in this paper is on device connectivity management which allows connectivity observation and bearer selection.

Devices may be connected using cellular bearers, wireless bearers or may use wireline ones. A remote application on the server may observe line voltage and signal strength at the device side. For this purpose, the application establishes an observation relationship with the device in order to set the observation policy. The device sends periodic or triggered reports with requested information until the application cancels the observational relationship. The application may query about multiple parameters related to connectivity on the device, including used network bearer and available network bearers. If for example the device has cellular network connectivity and supports WLAN connectivity, and WLAN coverage is available, then the application may request bearer selection with preferred WLAN bearer.

OMA traps that may be used for connectivity management are Geographic trap, Received power trap, Call drop trap, QoS trap, and Data speed trap [17]. Geographic trap may be used for location based bearer selection. It goes to active when a device enters into a specific geographic area. Whenever the device leaves that specific geographic area, the Geo trap goes to inactive. The Received power trap may be used for bearer selection based on received signal strength at the device. Whenever a device's received power drops below an application-specified value (TrapActivePower), it causes this trap to go active. Alternatively, when received power rises above another application-specified value (TrapInactivePower), it causes this trap to go inactive. In cases when the trap goes active or inactive, the device notifies the application. The device can have several instances of this kind of trap to monitor various network types (e.g. WiFi, WCDMA, LTE etc). The application may observe the call drops in a predefined period of time. If the device exposes QoS metrics functionality, then the application may observe the received QoS at the device side using the QoS trap. The Data speed trap triggers whenever an average data speed reaches certain threshold value.

III. DEVICE CONNECTIVITY MANAGEMENT MODELS

We model the device state in the context of device connectivity management. Fig.1 shows the device connectivity management model.

In *disconnected* state, the device is not connected to the network. When the device is switched on it registers with the network and becomes connected. In *connected* state, the device may be queried about its location and its connectivity parameters. When the signal strength of the used bearer drops, the Received power trap becomes active and the device moves to *marginal* state. In *marginal* state, if the signal strength rises, the Received power trap becomes inactive and the device moves to *connected* state, or the device may change bearer and move to connected state. In *connected* or *marginal* state, the device may enter or exit a predefined area, which results in Geo trap activation or deactivation respectively. While being in *connected* or *marginal* state, the device may be disconnected by the application or switched off.

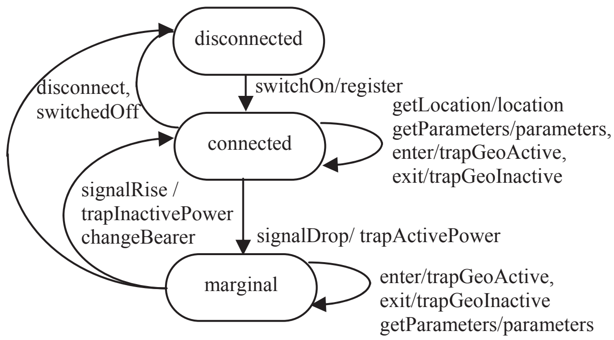

Fig. 1 Device connectivity management state model

We use the mathematical formalism of Labeled Transition Systems (LTS) to formally describe the model [18].

By $CM_D = (S_D, Inp_D, \rightarrow_D, s^0_D)$ it is denoted an LTS representing the Device's application view on the connectivity management state where:

S_D = { Disconnected [s_1^D], Connected [s_2^D], Marginal[s_3^D]};

Inp_D = { switchOn [t_1^D], getLocation [t_2^D], getParameters[t_3^D], signalDrop[t_4^D], enter[t_5^D], exit[t_6^D], signalRise[t_7^D], changeBearer[t_8^D], switchOff[t_9^D], disconnect[t_{10}^D]};

$\rightarrow_D$ = { ($s_1^D\ t_1^D\ s_2^D$), ($s_2^D\ t_2^D\ s_2^D$), ($s_2^D\ t_3^D\ s_2^D$), ($s_2^D\ t_5^D\ s_2^D$), ($s_2^D\ t_6^D\ s_2^D$), ($s_2^D\ t_4^D\ s_3^D$), ($s_3^D\ t_5^D\ s_3^D$),($s_3^D\ t_6^D\ s_3^D$), ($s_3^D\ t_3^D\ s_3^D$), ($s_3^D\ t_7^D\ s_2^D$), ($s_3^D\ t_8^D\ s_2^D$), ($s_3^D\ t_9^D\ s_1^D$), ($s_3^D\ t_{10}^D\ s_1^D$),($s_2^D\ t_9^D\ s_1^D$), ($s_2^D\ t_{10}^D\ s_1^D$)}

s^0_D = { s_1^D }.

Short notations of states' and inputs' names are given in brackets.

Using trap mechanism, different MEC applications which add value to device connectivity management may be designed.

Fig.2 shows the device connectivity management model as seen by an MEC application which applies location based bearer selection logic. The Location-based Bearer Selection (LBS) application assumes that there is a predefined geographic area in which a preferred bearer is used. For example for the university campus area with full Wi-Fi coverage, the preferred bearer is Wi-Fi.

By $CM_{App}= (S_{App}, Inp_{App}, \rightarrow_{App}, s^0_{App})$ it is denoted an LTS representing the MEC application's view on the connectivity management state where:

S_{App} = { AppDisconnected [s_1^A], AppConnected [s_2^A], ConnectedInArea[s_3^A], ConnectedOutArea[s_4^A], ConnectedInAreaPreffered[s_5^A], ConnectedInAreaNotPreffered[s_6^A], AppMarginal[s_7^A], BadSignal[s_8^A],};

Inp_{App} = { register[t_1^A], $location_{InArea}$ [t_2^A], $parameters_{UsedPrefered}$[$t_3^A$], trapActivePower[$t_4^A$], trapGeoActive[$t_5^A$], trapGeoInactive[$t_6^A$], trapInactivePower[$t_7^A$], timerExpiry[$t_8^A$], $parameters_{PreferredAvailable}$[$t_9^A$], $parameters_{HasAvailable}$[$t_{10}^A$], $parameters_{NoAvailable}$[t_{11}^A], deregister[t_{12}^D], disconnect[t_{13}^D], $location_{OutArea}$[t_{14}^A], $parameters_{PreferredUnavailable}$[$t_{15}^A$], };

$\rightarrow_{App}$ = { $(s_1^A\ t_1^A\ s_2^A)$, $(s_2^A\ t_2^A\ s_3^A)$, $(s_2^A\ t_{14}^A\ s_4^A)$, $(s_3^A\ t_3^A\ s_5^A)$, $(s_3^A\ t_9^A\ s_5^A)$, $(s_3^A\ t_{15}^A\ s_6^A)$, $(s_3^A\ t_6^A\ s_4^A)$, $(s_4^A\ t_5^A\ s_3^A)$, $(s_2^A\ t_4^A\ s_7^A)$, $(s_7^A\ t_6^A\ s_7^A)$, $(s_7^A\ t_5^A\ s_7^A)$, $(s_7^A\ t_7^A\ s_2^A)$, $(s_7^A\ t_8^A\ s_8^A)$, $(s_8^A\ t_{10}^A\ s_2^A)$, $(s_8^A\ t_{11}^A\ s_1^A)$, $(s_2^A\ t_{12}^A\ s_1^A)$, $(s_2^A\ t_{13}^A\ s_1^A)$, $(s_7^A\ t_{12}^A\ s_1^A)$, $(s_7^A\ t_{13}^A\ s_1^A)$, $(s_8^A\ t_{12}^A\ s_1^A)$, $(s_8^A\ t_{13}^A\ s_1^A)$}

s^0_{App} = { s_1^A }.

We use the concept of weak bisimulation to formally verify the suggested models [19].

Proposition: The labeled transition systems CM_{App} and CM_D are weakly bisimilar.

Proof: As to definition of weak bisimulation, provided in [18], it is necessary to identify a bisimilar relation between the states of both LTSs and to identify respective matching between transitions.

Let us denote by U_{AppD} the following relation between CM_{App} and CM_D where U_{AppD}={(s_1^D, s_1^A), (s_2^D, s_2^A), (s_3^D, s_7^A),}. Then, for the following network events, we identify the respective transitions between states of CM_{App} and CM_D:

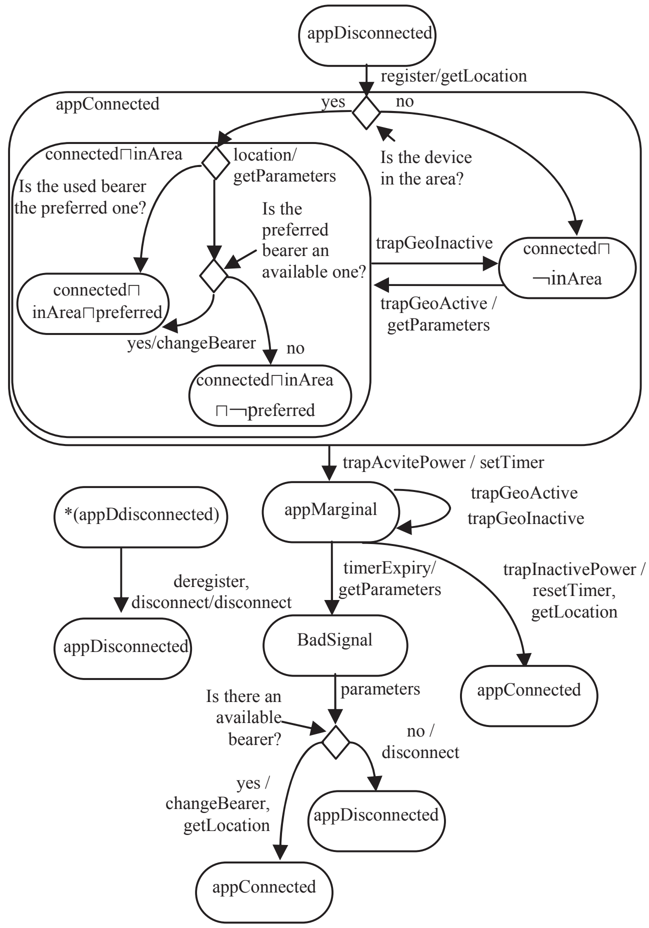

Fig. 2 Location-based device connectivity management state model

1) On device registration: for $(s_1^D\ t_1^D\ s_2^D) \exists (s_1^A\ t_1^A\ s_2^A)$.
2) The device is in the predefined area and it uses the preferred bearer: for $(s_2^D\ t_2^D\ s_2^D) \exists (s_2^A\ t_2^A\ s_3^A)$ and $(s_3^A\ t_3^A\ s_5^A)$.
3) The device is in the predefined area and the preferred bearer is not used but available: for $(s_2^D\ t_2^D\ s_2^D) \exists$ $(s_2^A\ t_2^A\ s_3^A)$ and $(s_3^A\ t_9^A\ s_5^A)$.
4) The device is in the predefined area and the preferred bearer is not available: for $(s_2^D\ t_2^D\ s_2^D) \exists (s_2^A\ t_2^A\ s_3^A)$ and $(s_3^A\ t_{15}^A\ s_6^A)$.
5) The device is out of the predefined area: for $(s_2^D\ t_2^D\ s_2^D) \exists (s_2^A\ t_{14}^A\ s_4^A)$.
6) The device exits the predefined area: for $(s_2^D\ t_6^D\ s_2^D) \exists$ $(s_3^A\ t_6^A\ s_4^A)$.

7) The device enters the predefined area: $(s_3^D\ t_5^D\ s_3^D) \exists (s_4^A\ t_5^A\ s_3^A)$.
8) The signal strength of the used bearer drops: $(s_2^D\ t_4^D\ s_3^D) \exists (s_2^A\ t_4^A\ s_7^A)$
9) The signal of the used bearer rises: $(s_3^D\ t_7^D\ s_2^D) \exists (s_7^A\ t_7^A\ s_2^A)$
10) The signal strength of the used bearer is low and there is another available bearer. The application initiates bearer change: $(s_3^D\ t_8^D\ s_2^D) \exists (s_7^A\ t_8^A\ s_8^A)$ and $(s_8^A\ t_{10}^A\ s_2^A)$.
11) The signal strength of the used bearer is low and there is no available bearer. The application requests the device to disconnect: $(s_3^D\ t_{10}^D\ s_1^D) \exists (s_7^A\ t_8^A\ s_8^A)$ and $(s_8^A\ t_{13}^A\ s_1^A)$.
12) While the device is connected, it may be switched off: $(s_2^D\ t_9^D\ s_1^D) \exists (s_2^A\ t_{12}^A\ s_1^A)$.
13) While the signal strength of the used bearer is low, the device may be switched off: $(s_3^D\ t_9^D\ s_1^D) \exists (s_7^A\ t_{12}^A\ s_1^A)$ and $(s_7^A\ t_8^A\ s_8^A)$, $(s_8^A\ t_{12}^A\ s_1^A)$.
14) While the device is connected, the application may request the device to disconnect: $(s_2^D\ t_{10}^D\ s_1^D) \exists (s_2^A\ t_{13}^A\ s_1^A)$.
15) While the signal strength of the used bearer is low, the application may request the device to disconnect: $(s_3^D\ t_{10}^D\ s_1^D) \exists (s_7^A\ t_{12}^A\ s_1^A)$ and $(s_7^A\ t_8^A\ s_8^A)$, $(s_8^A\ t_{13}^A\ s_1^A)$.

Following the same approach other MEC applications related to connectivity management may be designed. The applications may apply policies for device bearer selection using Call drop trap, QoS trap and Data speed trap.

There are also events related to Policy and Charging Control (PCC) that may trigger a bearer change [20]. Such events include e.g. out of credit (credit is no longer available), usage report, enforcement of Application Detection Control rule, etc.

In the next section, we present a method for automatic detection of undesired interaction between applications. The method considers application interaction as a satisfiability problem.

III. Connectivity management applications

Description logic is a formal language used for knowledge represantions and reasoning about it [21]. The basic syntactic blocks used to represent the knowledge base are atomic concepts, atomic roles and constants. The basic components of the knowledge base are Terminology box (TBox) which introduces terminology in the application domain and the Assesrtion box (ABox) which contains assertions about constants. Typical reasoning on knowledge base is to determine whether a description is non-contradictory or whether given description subsumes another one.

A. Semantic annotation for device connectivity management

Our approach to definition of atomic concepts is to represent the device states and bearer related facts in the CM model as concepts.

Let us assume that there is a finite set of bearer indices which represent the possible bearers that may be used by a particular device. The following concepts are defined:

disconnected, device is disconnected;
connected$_b$, device is connected using bear *b*;
marginal$_b$, device's received power of used bearer is below an application-specified value;
badSignal, device needs to change the used bearer;
inArea, device is in a predefined area;
preferred$_b$, bearer *b* is preferred bearer in the area;
qosAcceptable$_b$, quality of service of bearer *b* is acceptable;
available$_b$, bear *b* is available;
unavailable, there are no available bearers.

The transitions that change the device state are defined as atomic roles:

signalDrop, received power of used bearer drops below application-specified value;
signalRise, received power of used bearer rises above application-specified value;
enter, device enters the predefined area;
exit, device exits the predefined area;
qosDecrease, QoS of used bearer becomes unacceptable;
timerExpiry, time guarded hysteresis of the received power is over;
connect, device connects to the network
disconnect, device disconnects from the network

Our terminology box contains expressions showing the changes in CM model and statements specifying the relationship between the events that cause transitions.

$$disconnected \sqcap available_b \sqsubseteq \exists connect.connected_b \quad (1)$$

$$connected_b \sqsubseteq \exists signalDrop.marginal_b \quad (2)$$

$$marginal_b \sqsubseteq \exists signalRise.connected_b \quad (3)$$

$$marginal_b \sqsubseteq \exists timerExpiry.badSignal \quad (4)$$

$$badSignal \sqcap available_b \sqsubseteq \exists connect.connected_b \quad (5)$$

$$badSignal \sqcap unavailable \sqsubseteq \exists disconnect.disconnected \quad (6)$$

$$connected_b \sqsubseteq \exists disconnect.disconnected \quad (7)$$

We need expressions that describe the device mobility which is based on OMA Geo trap:

$$\neg inArea \sqsubseteq \exists enter.inArea \sqcap preferred_b \quad (8)$$

$$inArea \sqcap preferred_b \sqsubseteq \exists exit.\neg inArea \quad (9)$$

Let us denote by DEV the set of all devices. By CMS we denote the states s_i in the CM model. The assertion box contains one statement presenting the initial state for each device:

$$s_0 : \sqcap_{d \in DEV} (disconnected \sqcap available_b \sqcap inArea \sqcap preferred_b \sqcup disconnected \sqcap \neg available_b \sqcap available_c \sqcap inArea \sqcap preferred_b \sqcup disconnected \sqcap \neg inArea \sqcap available_c \sqcup disconnected \sqcap unavailable).$$

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

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

The device state changes by means of actions defined as action functions. An action function $Func_{CMS}$ for given state corresponds to the possible transitions in the CM model. For example, the expression $Func_{CMS}(connected_b) = \{signalDrop\} \cup \{disconnect\} \cup \{enter\} \cup \{exit\}$ means that if the device is connected, the signal strength of the used bearer may drop, the device may disconnect, enter or exit the predefined area.

The fact that each device can change the CM state only by means of certain actions is represented by the following statement: for all $s \in CMS$, and all $R \notin Func_{CMS}(s)$, $s \sqsubseteq \forall R.s$.

Services are modeled as transformations on the knowledge base using contexts $C[\varphi]$ as subformula φ of any formula ψ.

B. Location-based bearer selection

The refinement for LBS application is defined by the following statements:

$$C_1[LBS \sqcap disconnected \sqcap available_b \sqcap inArea \sqcap preferred_b] \sqsubseteq \exists connect.C_2[connected_b \sqcap inArea \sqcap preferred_b] \quad (10)$$

$$C_3[\neg LBS \sqcap disconnected \sqcap available_c \sqcap inArea \sqcap preferred_b] \sqsubseteq \exists connect.C_4[connected_c \sqcap inArea \sqcap preferred_b] \quad (11)$$

$$C_5[LBS \sqcap disconnected \sqcap \neg available_b \sqcap inArea \sqcap preferred_b] \sqsubseteq C_6[disconnected] \quad (12)$$

$$C_7[LBS \sqcap connected_b \sqcap available_c \sqcap inArea \sqcap preferred_c \sqsubseteq \exists connect.C_8[connected_c] \quad (13)$$

$$C_9[\neg LBS \sqcap connected_b \sqcap available_c \sqcap inArea \sqcap preferred_c \sqsubseteq C_{10}[connected_b] \quad (14)$$

$$C_{11}[LBS \sqcap connected_b \sqcap \neg available_c \sqcap inArea \sqcap preferred_c] \sqsubseteq \exists disconnect.C_{12}[disconnected] \quad (15)$$

$$LBS \sqsubseteq \neg(connected_b \sqcap inArea \sqcap available_c \sqcap preferred_c) \quad (16)$$

C. Quality of service based bearer selection

The Quality of Service-based Bearer Selection (QBS) service requires bearer change if the QoS available on the used bearer decreases under predefined value. The refinement for QBS application is defined by the following statements:

$$C_1[\neg QBS \sqcap connected_b] \sqsubseteq \exists qosDecrease_b.C_2[connected_b] \quad (17)$$

$$C_3[QBS \sqcap connected_b] \sqsubseteq \exists qosDecrease_b.C_4[connected_b \sqcap qosUnacceptable_b] \quad (18)$$

$$C_5[QBS \sqcap connected_b \sqcap qosUnacceptable_b \sqcap available_c] \sqsubseteq \exists connect.C_6[connected_c] \quad (19)$$

$$C_7[QBS \sqcap connected_b \sqcap qosUnacceptable_b \sqcap unavailable] \sqsubseteq \exists disconnect.C_8[disconnected] \quad (20)$$

$$QBS \sqsubseteq \neg(connected_c \sqcap qosUncceptable_c) \quad (21)$$

Possible feature interaction may occur when the device is in the predefined area and the QoS available on used preferred bearer decreases.

V. Reasoning on feature interaction

When introducing new services, it is important to find out whether a new service is contradictory to existing concepts i.e. whether it is satisfiable or unsatisfiable with respects of axioms in TBox representing the CM model.

A. Tableau method

We use a tableau method defined in [21]. The tableau $t \stackrel{\text{def}}{=} \{ \langle b \mid p : C \rangle \}$ is a set of prefixed formulae where the prefix of given formula is consisted of a binary string $b := \varepsilon \mid (1|0)^+$ and a string of alternating names $p := n(Rm)^+$, and C is concept. Here ε is the empty string, n and m are names of individuals, R stands for the names of roles, and $()^+$ denotes one or more occurrences. The tableau method is shown in Table I.

TABLE I. Tableau Method

AND: $\dfrac{\langle b \mid p : C \cap D \rangle}{\langle b \mid p : C \rangle \quad \langle b \mid p : D \rangle}$	
OR: $\dfrac{\langle b \mid p : C \cup D \rangle}{\langle b_M 0 \mid p : C \rangle \quad \langle b_M 1 \mid p : D \rangle}$	b_M maximal for b
SOME: $\dfrac{\langle b \mid p : \exists R.C \rangle}{\langle b \mid pRn : C \rangle}$	pRn new (unless pR exists in the branch)
KB: $\dfrac{\vdots}{\langle b \mid p : \neg C \cup D \rangle}$	p present in b and $C \sqsubseteq D \in T$

B. Detection of interaction between LBS and QBS

The tableau algorithm for detecting interactions between LBS and QBS services proceeds as follows:

Applying AND to the start formula produces four cases:

$\langle \varepsilon \mid s_0: \sqcap_{d\in DEVICES}$ $(disconnected \sqcap available_b \sqcap inArea \sqcap preferred_b \sqcup disconnected \sqcap \neg available_b \sqcap available_c \sqcap inArea \sqcap preferred_b \sqcup disconnected \sqcap \neg inArea \sqcap available_c \sqcup disconnected \sqcap unavailable) \rangle$

1. In case of $disconnected \sqcap available_b \sqcap inArea_a \sqcap preferred_b$

1.1Applying KB to rule (10) produces

$\langle \varepsilon \mid s_0: \neg disconnected \sqcup \neg available_b \sqcup \neg\ inArea \sqcup \neg\ preferred_b \sqcup \exists available_b.(connected_b \sqcap qosAcceptable_b)\rangle$

1.2. Applying OR gives two branches:

1.2.1 $\langle 0 \mid s_0: \neg disconnected\rangle$ which is closed because of the appearance of $\langle 0 \mid s_0 : disconnected\rangle$ in this segment earlier.

1.2.2 $\langle 0 \mid s_0: \neg available_b\rangle$ (closed).

1.2.3 $\langle 0 \mid s_0: \neg inArea\rangle$ (closed).

1.2.4 $\langle 0 \mid s_0: \neg\ preferred_b\rangle$ (closed).

1.2.5 $\langle 1 \mid s_0: \exists connect_b.(connected_b \sqcap inArea)$

1.3 Applying SOME gives $\langle 1 \mid s_0\ connect\ s_1: (connected_b \sqcap inArea \sqcap preferred_b) \rangle$

1.4 We derive rule (18) and applying KB produces

1.4.1 $\langle 1 \mid s_0\ connect\ s_1: \neg(connected_b \sqcap inArea \sqcap preferred_b) \sqcup \exists qosDecrease_b.(connected_b \sqcap qosUnacceptable_b \sqcap preferred_b) \rangle$ and after applying OR

1.4.2 $\langle 10 \mid s_0\ connect\ s_1: \neg(connected_b \sqcap inArea \sqcap preferred_b)\rangle$ (closed)

1.4.3 $\langle 11 \mid s_0\ connect\ s_1: \exists qosDecrease_b.(connected_b \sqcap qosUnacceptable_b \sqcap inArea \sqcap preferred_b)\rangle$ and applying SOME

$\langle 11 \mid s_0\ connect\ s_1\ qosDecrease_b\ s_2: connected_b \sqcap qosUnacceptable_b \sqcap inArea \sqcap preferred_b) \rangle$

1.5. Subsequent derivation is the rule (19) for which we apply KB and the result is

1.5.1 $\langle 110 \mid s_0\ connect\ s_1\ qosDecrease_b\ s_2: \neg connected_b \sqcap qosUnacceptable_b \sqcap inArea \sqcap preferred_b \rangle$ (closed)

1.5.2 $\langle 111 \mid s_0\ connect\ s_1\ qosDecrease_b\ s_2: \exists connect.connected_c \sqcap inArea \sqcap preferred_b\rangle$ and after applying SOME it produces $\langle 111 \mid s_0\ connect\ s_1\ qosDecrease_b\ s_2\ connect\ s_3: connected_c \sqcap inArea \sqcap preferred_b\rangle$ which contradicts to $LBS \sqsubseteq \neg(connected_c \sqcap inArea \sqcap preferred_b)$.

2. In case of $disconnected \sqcap \neg available_b \sqcap available_c \sqcap inArea \sqcap preferred_b$ the device remains disconnected as to rule (12).

3. In case of $disconnected \sqcap \neg inArea \sqcap available_c$

3.1 Applying KB to the rule (1) and eliminating the closed cases gives

3.1.1 $\langle 1 \mid s_0: \exists connect_c.(connected_c \sqcap \neg inArea) \rangle$ to which applying SOME results in $\langle 1 \mid s_0\ connect\ s_1: connected_c\ \neg inArea) \rangle$

3.2 We derive rule (8) and applying consecutively KB, OR and SOME gives two branches

3.2.1 $\langle 1 \mid s_0\ connect\ s_1\ enter\ connected_c \sqcap inArea \sqcap preferred_b \sqcap available_b) \rangle$ for which we derive rule (13) and again applying KB, OR and SOME produces

$\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ connect\ s_3: connected_b \sqcap inArea \sqcap preferred_b \sqcap available_b) \rangle$

3.2.2 $\langle 1 \mid s_0\ connect\ s_1\ enter\ connected_c \sqcap inArea \sqcap preferred_b \sqcap \neg available_b) \rangle$ for which we derive $connected_b \sqcap \neg available_c \sqcap inArea \sqcap preferred_c \sqsubseteq \exists disconnect.disconnected$ and applying KB, OR and SOME produces $\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ disconnect\ s_3: disconnected\rangle$

3.3 To $\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ connect\ s_3: connected_b \sqcap inArea \sqcap preferred_b \sqcap available_b) \rangle$ we apply the similar steps as those in (1.5.1) and it gives $\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ connect\ s_3\ qosDecrease_b\ s_4\ connect\ s_5: connected_c \sqcap inArea \sqcap preferred_b\rangle$ which contradicts to LBS.

4. In case of *disconnected⊓unavailable,* the device remains disconnected.

The result is closed tableau which means that $\delta_{QBS}(\delta_{LBS}(CMS))$ interacts on activation $\{QBS\} \cup \{LBS\}$.

It is important to mention that the feature interaction can be detected automatically since the programmability of the algorithm. Using the semantic annotation, the connectivity management applications may be described by Ontology Web Language (OWL), where the concepts are represented by classes, the roles are described as restrictions. The algorithm for detection of feature interaction may be automated by OWL reasoners that deduce implicit or explicit knowledge.

V. CONCLUSION

Connectivity management includes M2M connection provisioning, management and analysis across cellular and wireless networks. Applications devoted to connectivity management need to control in real-time the device connectivity responding to dynamic changes in radio conditions. Application deployment in MEC environment can reduce latency and improve the usage of radio resources.

In this paper, we propose device connectivity management models that are based on the trap diagnostics and monitoring mechanism defined by OMA. Based on real-time exchange of information about device connectivity parameter values, MEC applications may monitor and control the cellular or wireless bearers used by M2M devices. Models are formally described and verified.

The synthesis of device connectivity models is based on semantic annotation. This annotation is used to construct the knowledge base which formally describes the application logic adding value to connectivity management. Different policies may be applied to connectivity management which may result in unexpected or even undesired feature interaction which calls for a tool for detecting such issues in advance.

We propose a method for formal description of applications for M2M connectivity management and an approach to feature interaction detection. Once detected at the specification phase, feature interactions may be avoided by applying policies and rules.

The presented results outline a possible solution and the approach seems to be promising as far as the scalability is achievable because of algorithm's programmability.

ACKNOWLEDGMENT

The research is conducted under the grant of project DH07/10-2016, funded by National Science Fund, Ministry of Education and Science, Bulgaria.

REFERENCES

[1] A. Ahmed, E. Ahmed, "A survey on Mobile Edge Computing," 10th IEEE International Conference on Intelligent Systems and Control (ISCO 2016), 2016, pp. 1-8.

[2] L. Gupta, R. Jain, H. A. Chan, "Mobile Edge Computing – an important ingredient of 5G networks," IEEE Software Defined Networks, Newsletter, March 2016, Available at: http://sdn.ieee.org/newsletter/ march-2016/mobile-edge-computing-an-important-ingredient-of-5g-networks

[3] Y. Chen, L, Ruchenbusch, "Mobile Edge Computing: brings the value back to networks," *IEEE Software Defined Networks*, Newsletter, March 2016, Available at: http://sdn.ieee.org/newsletter/march-2016/mobile-edge-computing-bring-the-values-back-to-networks

[4] L. Latvakoski, et al. "A survey on M2M service networks", Computers, 2014, vo.3, pp.130-173, doi: 10.3390/computers3040130

[5] C.S. Shih. C. T. Chou, K. J. Lin, B. L. Tsai, C. H Lee, D. Cheng, C. J. Chou "Out-of-Box device management for large scale cyber-physical systems," IEEE International Conferences iThings, GreenCom, and CPSCom, 2014, pp.402 – 407.

[6] S. Datta, C. Bonnet, "A lightweight framework for efficient M2M device management in oneM2M architecture," International Conference on Recent Advances in Internet of Things (RIoT), 2015, pp.1-6.

[7] C. Pereira, A. Aguiar, "Towards efficient mobile M2M communi-cations: Survey and Open Challenges," *Sensors,* 2014, no. 14, pp.19582-19608.

[8] C. Maternaghan, K. Turner, "Policy conflicts in home automation," *Computer Networks*, vol.57 issue 12, pp.2429-2241.

[9] A.L.Dominguez. "Detection of feature interactions in automotive active safety features," PhD thesis, School of Computer Science, University of Waterloo, 2012.

[10] Y.B.Lin, et al. "EasyConnect: A management system for IoT devices and its applications for interactive design and art," *IEEE Internet of Things*, vol.2, issue 6, pp551-561.

[11] P. Zave. "Modularity in distributed feature composition. In Software requirements and design: the work of Michael Jackson," 2010, pp. 267–290. Good Friends Publishing,

[12] S. Apel, C. Kastner, B. Garvin. "Exploring feature interactions in the wild: the new feature interaction challenge," FOSD,ACM, 2013, pp.1-8.

[13] Pencheva, E., I. Atanasov. **"**Detection of CAMEL feature interaction", *International Journal on Information Technology and Security*, 2010, (1), pp. 25-42.

[14] Atanasov, I., E. Pencheva. "CAMEL service interaction detection", *International Journal on Information Technologies and Control*, 2010, (4), pp.2-9.

[15] Atanasov, I., E. Pencheva. "A formal approach to service interaction detection in mobile networks". Proc. of 10th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems (SEPADS '11) Cambridge, UK, 2011, pp.118-123

[16] Atanasov, I., E. Pencheva. "Reasoning on Service Interaction in Mobile Networks", *International Journal of Computers and Communications*, 2011, vol.5 (2), pp.59-66.

[17] Open Mobile Alliance, Diagnostics and monitoring trap events specifications, 2013, OMA-TS-DiagMonTrapEvents-V1_2-20131008-A

[18] A. Chebieb and Y. A. Ameur, "Formal Verification of Plastic User Interfaces Exploiting Domain Ontologies," *2015 International Symposium on Theoretical Aspects of Software Engineering*, Nanjing, 2015, pp. 79-86.

[19] D. Escrig, J. Keiren, T. Willemse, "Games for Bisimulations and Abstraction", Cornel University Library, arXiv:1611.00401 [cs.LO]

[20] 3GPP TS 23.203 Policy and charging control architecture, Release 13, v13.7.0, 2016.

[21] F. Baader, U. Sattler, "Tableau Algorithms for Description Logics," Chapter in Automated Reasoning with Analytic Tableaux and Related Methods, vol.1847, Lecture Notes for Description Logics, pp.1-18.