ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ
Факултет по Телекомуникации
Катедра Комуникационни мрежи

Маг. инж. Анастас Николов Николов

ИЗСЛЕДВАНЕ НА МОДЕЛИ И МЕТОДИ ЗА АВТОНОМЕН МЕНИДЖМЪНТ В ИНТЕРНЕТ НА НЕЩАТА

А В Т О Р Е Ф Е Р А Т

на дисертация за придобиване на образователна и научна степен
"ДОКТОР"

Област: 5. Технически науки

Професионално направление: 5.3 Комуникационна и компютърна техника

Научна специалност: Комуникационни мрежи и системи

Научен ръководител: проф. дтн Евелина Пенчева

СОФИЯ, 2018 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Комуникационни мрежи“ към Факултет по Телекомуникации на ТУ-София на редовно заседание, проведено на 04.12.2017 г..

Публичната защита на дисертационния труд ще се състои на 29.03.2018 г. от 13 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед № ОЖ-377 / 20.12-2017 г. на Ректора на ТУ-София в състав:

1. Проф. д.т.н. Евелина Пенчева – председател
2. Доц. д-р Венцислав Трифонов – научен секретар
3. Проф. д-р Борис Цанков
4. Проф. д.т.н. Михаил Михов
5. Доц. д-р Пенчо Пенчев

Рецензенти:

1. Проф. д-р Борис Цанков
2. Доц. д-р Венцислав Трифонов

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет по Телекомуникации на ТУ-София, блок №1, кабинет № 1254.

Дисертантът е редовен докторант към катедра „Комуникационни мрежи“ на Факултет по Телекомуникации. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Анастас Николов

Заглавие: Изследване на модели и методи за автономен мениджмънт в Интернет на нещата

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Предизвикателствата, пред които се изправя съвременният Интернет, потенциалните изисквания и нужните технически средства, очертават критични направления, в които да се реализират редица научни изследвания. Съставя се нова обща представа за Интернет, която се оформя в понятието Интернет на бъдещето. Интернет на бъдещето реализира идеята за бързи и адаптивни мрежи, които разполагат със способността да посрещнат нуждите на потребители и устройства. Обединяването на множество автономни свойства с цел да се поддържа ниско ниво на оперативните разходи, докато самите мрежи се разгръщат в огромни мащаби, е част от идеята на мрежовите инфраструктури от ново поколение. Безусловно подобен тип свойства биха били: самоконфигуриране, самовъзстановяване, самооптимизация и самозащита. Проучванията върху познатите ни биологични апарати са с пряко влияние върху изследванията, свързани с автономните изчислителни системи. Целта е да се разработят системи, които биха могли самостоятелно да взимат решения, самоуправлявайки се, в случай, че се изправят срещу една или редица сложни задачи. Базираните на политики системи, проактивните мрежи и софтуерните агенти дават възможност за автоматизиране на управлението на задачи, намаляването на разходите за осъществяването му и намаляване на времето за реакция. Една система може да се самоконфигурира, като следва определени политики от високо ниво и се адаптира към настъпили промени. Поредица от явления водят със себе си новите предизвикателства в областта на бизнеса и технологиите: сходимост на услуги, сходимост на информационни и комуникационни технологии, огромно различие от цифрови услуги. Създалите се обстоятелства предразполагат мрежите и управлението им да функционират с ограничена намеса от страна на човека, да се развие интелигентност, сигурност и автономност. В настоящия дисертационен труд е направен опит за решаване на някои от проблемите, свързани с изграждането на автономни системи в Интернет на бъдещето.

Цел на дисертационния труд, основни задачи и методи за изследване

Основната цел на дисертационния труд е да се изследват възможностите за автономен мениджмънт в комуникации от машинен тип и да се предложат нови модели и методи за самоконфигуриране и самовъзстановяване на устройства и приложения в Интернет на бъдещето.

Основната цел е декомпозирана на следните специфични подцели, за всяка от които са формулирани задачи:

1. Изследване на функции за автономен мениджмънт в Интернет на нещата.
2. Изследване на функции за автономен мениджмънт в енергийни системи.
3. Изследване на модели и методи за самоконфигуриране на услуги в Интернет на нещата.

Математически понятия за детерминирани крайни автомати като системи с именувани преходи и биподобие са използвани за формална верификация на моделите за мениджмънт на конфигурацията и мениджмънт на работоспособността. За моделиране на автономно поведение са използвани интелигентни агенти, чиито основни цели са избор на носещ ресурс и управление на качеството на обслужване. Синтезирането на абстракции като процес на анотиране на ресурси

със семантични данни е използвано в областта на домашни интелигентни енергийни системи. Изследван е метод за самоконфигурирането на услуги в комуникации от машинен тип, който включва моделиране и формално описание на знанието чрез дескриптивна логика. Откриването на нежелани взаимодействия между услуги се базира на откриване на противоречия в базата знания при добавяне на нова услуга и може да се отстрани чрез прилагане на политики, свързани с приоритети на услуги.

Научна новост

Изследвани са методи за мениджмънт на устройства. Проектирани са модели за регистриране на устройства, мениджмънт на свързаността, мениджмънт на неизправности и управление на качеството на обслужване. Моделите са описани формално и са верифицирани. Предимствата на използването на предложените модели в сравнение със съществуващите, специфични такива са, че са базирани на широко приета рамка за мениджмънт, която е технологично независима и процедурите по мениджмънт могат да бъдат автоматизирани и могат да се използват при разработване на хоризонтални платформи. Предложен е структурен метод за синтезиране на семантични данни в областта на домашни системи за енергийна ефективност. Методът включва моделиране на данни и дефинирането на интерфейси за уеб услуги. Семантичната анотация на данните дава знание за тяхната организация и съдържание и се използва при изграждане на автономни системи. Изследвани са модели и методи са самоконфигуриране на услуги. Автономните функции са свързани с автоматично откриване и разрешаване на нежелани взаимодействия между услуги. Разработен е метод за откриване на нежелани взаимодействия между услуги. Методът се базира на формално описание на логиката на услуги чрез твърдения в база знания и търсене на противоречия при въвеждане на описание на нова услуга.

Практическа приложимост

Моделите за регистриране на устройства, мениджмънт на свързаността, мениджмънт на неизправности и управление на качеството на обслужване може да се използват в системи за мениджмънт на устройства, като по този начин да улеснят процесите при поддържане на жизнения цикъл на системата. Моделите и методите за автономни енергийни системи може да се за по-ефективно използване на енергийните ресурси и намаляване е приложим при автоматичното договаряне на услугите, предоставяни от различни доставчици на услуги с цел осигуряване на адаптивна непрекъснатост на услугите за крайните потребители.

Апробация

Апробацията на дисертационния труд е извършена пред Катедрен съвет на катедра Комуникационни мрежи, Факултет по Телекомуникации на 04.12.2017 г.

Публикации

Резултатите от изследванията в дисертационния труд са представени в общо 15 авторски публикации. Осем от публикациите са индексирани в референтните бази данни Scopus и Web of Science. Три от публикациите са в международни списания от клас А със SJR (SCImago Journal Rank) индикатор**.** Към момента на оформяне на

дисертационния труд са забелязани 4 цитирания на публикации в Scopus и Web of Science.

Структура и обем на дисертационния труд

Дисертационният труд е в обем от 183 страници, като включва увод, 4 глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо 84 литературни източници на латиница. Работата включва общо 48 фигури и 8 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Първа глава. Анализ на предизвикателствата пред автономния мениджмънт в Интернет на бъдещето

В първа глава са изследвани особеностите на автономния мениджмънт в Интернет на бъдещето. В авторски публикации **[A1]** и **[A2]** е поставена основата за анализа на предизвикателствата пред автономния/самоуправляващ се Интернет на бъдещето върху стандартите, дефиниращи изискванията и архитектурата на взаимодействащи си бъдещи мрежи, които предоставят множество услуги и се самоуправляват. Прегледът на научните работи в областта води до изводи, които очертават насоките на научните изследванията и позволяват формулиране на основната цел и задачи на дисертационния труд

Втора глава. Изследване на функции за автономен мениджмънт в Интернет на нещата

Възможността за управление и контрол на устройства от разстояние е от съществена важност за комуникациите от машинен тип (Machine-to-Machine - M2M). Големият брой от различни по вид свързани устройства, които взаимодействат със заобикалящата ги среда без човешка намеса, се явява сериозно предизвикателство пред мрежовите оператори. Изследванията във втора глава са свързани с динамиката в обмяната на информация между сървър и клиент, като се цели да бъдат предложени модели и методи, които да бъдат използвани в процеса на конструиране на платформи за мениджмънт на M2M устройства.

Синтезирани са основни модели за регистриране на свързани устройства от гледна точка на сървър и на клиент, които се основават на рамката за мениджмънт на устройства Leightweight M2M (LWM2M), осигурена от Open Mobile Alliance (OMA). Моделите са разширени с допълнителни функционалности, като е предложен и метод за формалната им верификация чрез използване на слабото биподобие, представен в авторска публикация **[A3].**

Чрез $R_S = (S_S, Inp_S, \rightarrow_S, S_S^0)$ е обозначена LTS (Label Transition System), която представя модел на състоянието на регистрация на устройство от гледна точка на LWM2M сървър, където:

$S_S = \{Unregistered_S, Operational_S\}$;

$Inp_S = \{reg_{REQ}, deReg_{REQ}\}$;

$\rightarrow_S$ = {($Unregistered_S$ reg_{REQ} $Operational_S$), ($Operational_S$ reg_{REQ} $Operational_S$),
($Operational_S$ $deReg_{REQ}$ $Unregistered_S$)};

S_S^0 = {$Unregistered_S$ }.

От гледна точка на LWM2M сървър едно устройство е в състояние $Unregistered_S$, когато не е регистрирано. В противен случай се приема, че устройството е в състояние $Operational_S$. Когато LWM2M сървър получи заявка за регистриране (reg_{REQ}), се предприема автентикация на устройството, като посредством изпращането на отговор (reg_{RES}) регистрацията бива приета и устройството вече се счита за регистрирано. При получаване на заявка за дерегистриране ($deReg_{REQ}$) LWM2M сървърът реагира и същевременно прекратява регистрацията на дадено устройство.

Чрез R_D = (S_D, Inp_D, $\rightarrow_D$, S_D^0) е обозначена LTS, която представя модел на състоянието на регистрация на устройство от гледна точка на LWM2M клиент, където:

S_D = {$Unregistered_D$, *Registering*, $Operational_D$};

Inp_D = {T_{REG}, reg_{RES}, *softOffline*};

$\rightarrow_D$ = {($Unregistered_D$ T_{REG} *Registering*), (*Registering* reg_{RES} $Operational_D$),
($Operational_D$ T_{REG} *Registering*), ($Operational_D$ *softOffline* $Unregistered_D$)};

S_D^0 = {$Unregistered_D$}.

От гледна точка на LWM2M клиент едно устройство може да бъде в състояние $Unregistered_D$, когато е извън употреба и следователно не е регистрирано. В състояние *Registering* устройството е в процес на регистрация. В състояние $Operational_D$ устройството е регистрирано. Когато регистрацията е успешна, устройството поддържа регистрационен таймер (T_{REG}). В случай че таймерът изтече, устройството обновява своята регистрация. Един път регистрирано, устройството може да получи заявка от тип *softOffline*, като в този случай LWM2M клиентът изпраща заявка за дерегистрация към сървъра и устройството преминава в състояние, в което не е регистрирано.

Твърдение 2.1 Системите с именувани преходи R_D и R_S изразяват бисимулация помежду си.

Доказателство 2.1 По дефиниция, за да се докаже, че две системи изразяват бисимулация помежду си, трябва да се идентифицира съответствие между техните състояния и преходи. Дефинира се следната релация на състояния:

U_{RS} = {($Unregistered_D$, $Unregistered_S$), ($Operational_D$, $Operational_S$)}.

В този случай за първата двойка състояния от U_{RS} може да се твърди, че е вярно:

$Unregistered_D \exists\ t_1$ = ($Unregistered_D$ T_{REG} *Registering*), t_2 = (*Registering* reg_{RES} $Operational_D$); $Unregistered_S \exists\ t_3$ = ($Unregistered_s$ reg_{REQ} $Operational_S$).

За втората двойка състояния съществуват следните съответстващи си преходи:

$Operational_D \exists\ t_4$ = ($Operational_D$ T_{REG} *Registering*), t_5 = (*Registering* reg_{RES} $Operational_D$); $Operational_S \exists\ t_6$ = ($Operational_S$ reg_{REQ} $Operational_S$);

и $Operational_D \exists\ t_7$ = ($Operational_D$ *softOffline* $Unregistered_D$); $Operational_S \exists\ t_8$ = ($Operational_S$ $deReg_{REQ}$ $Unregistered_S$).

От изразената релацията U_{RS}, съответните преходи и установяването на бисимулация между тях { (t_1 , t_2) , (t_3) }, { (t_4 , t_5) , (t_6) } и { (t_7) , (t_8) }, може да се съди, че R_S и R_D биват в състояние на слабо биподобие.■

Разширяването на основните модели с допълнителните функционалности бива осъществено чрез използването на предпоставки и следствия на дадено състояние.

Съществува LTS, $T = (S, Inp, \rightarrow, s_0)$, където следствие и предпоставка на състоянието $s \in S$ биват означени съответно с $Pos\ (s) = \{q | q \neq s,\ q \in S,\ s \rightarrow q\}$ и с $Pre\ (s) = \{q | q \neq s,\ q \in S,\ q \rightarrow s\}$.

Нека $T = (S, Inp, \rightarrow, s_0)$, $s \in S$. Разширението на s нека бъде означено с $E\ (s)$ и бъде дефинирано като следната LTS, $E\ (s) = (S', Inp', \rightarrow', s'_0)$, където:

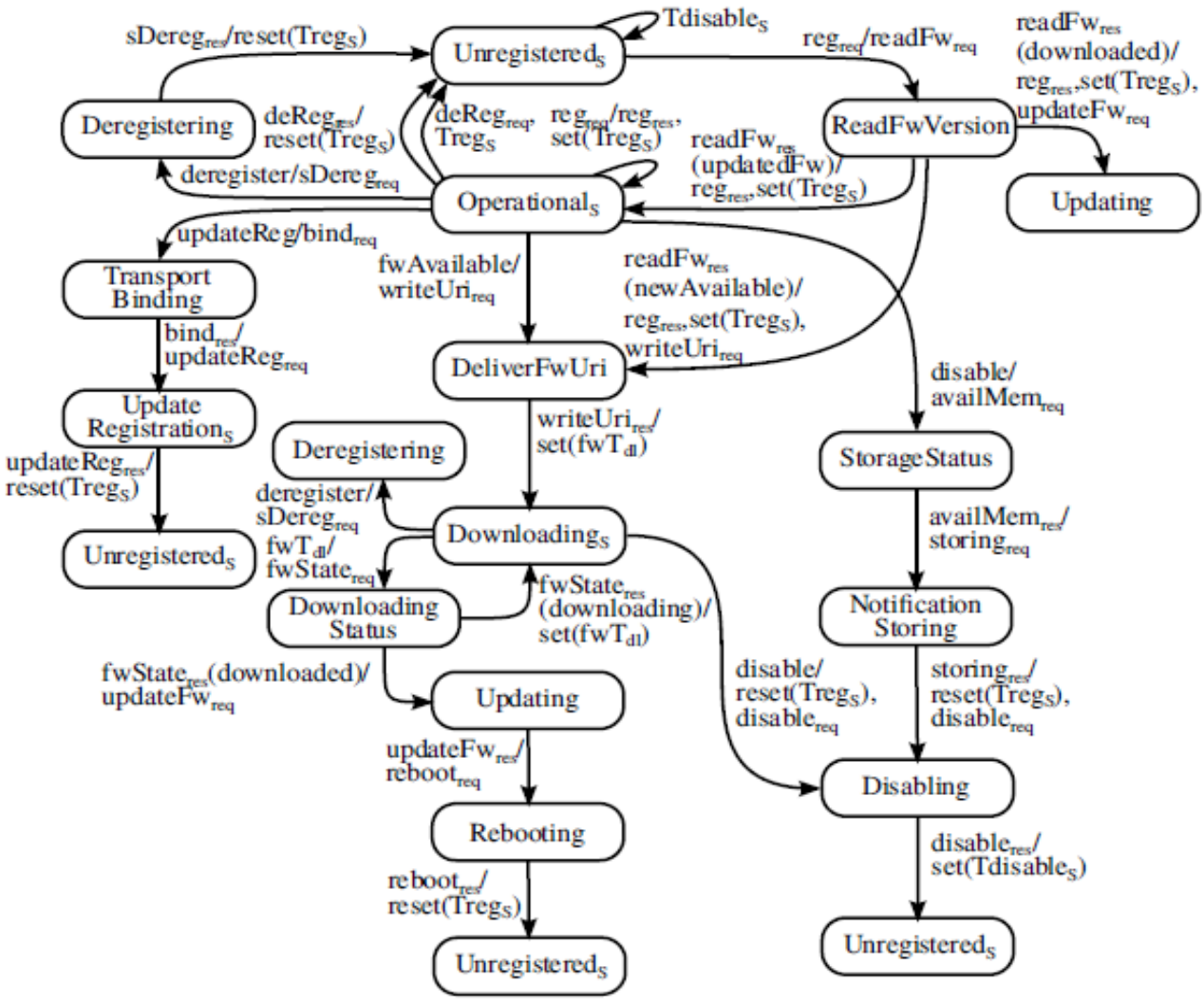
$S' \cap S = \varnothing$;

$\rightarrow' \cap \rightarrow = \{p \rightarrow q |\ (p \in Pre\ (s) \wedge q \in S') \vee (p \in S' \wedge q \in Pos\ (s))\}$;

$s'_0 = \{s' | s' \in S',\ Pre\ (s') \subseteq Pre\ (s)\}$;

$Inp' = \{a' | p \rightarrow q,\ p \in S',\ q \in S'\}$.

Разширените модели на регистриране от гледна точка на LWM2M сървър и клиент биват означени с R'_D и R'_S. Разширяването бива постигнато с добавяне на повече детайли към определени състояния в модела на устройството. Фиг. 2.1 представя разширения модел на състояние на регистрация на устройство от гледна точка на LWM2M сървър.



Фиг. 2.1 Разширен модел на състоянието на регистрацията на устройство от гледна точка на LWM2M сървър

Моделът, който представя гледната точка на сървър относно състоянието на регистрация на едно устройство с текуща софтуерна версия, бива разширен както следва:

Pre ($Operational_S$) = {*ReadFwVersion*} и

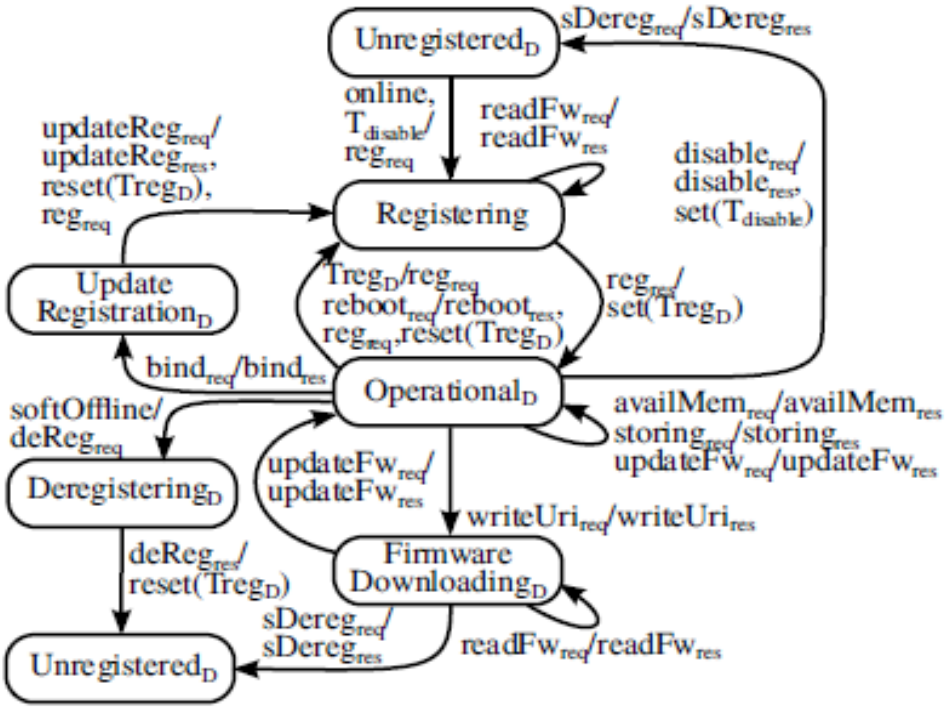
Pos ($Operational_S$) = {*Deregistering, MemoryStatus,* $UpdateRegistration_S$, *FwUpdate, NotificationStoring*$_S$, *Rebooting, DownloadingStatus, TransportBinding, Disabling, DeliverFwUri,* $FirmwareDownloading_S$}

Фиг. 2.2 изобразява модел на състоянието на регистрация на устройство от собствената му гледна точка.

Нека $R'_S = (S'_S, Inp'_S, \rightarrow'_S, s_S^0)$ бъде LTS, която изразява гледната точка на сървър относно състоянието на регистрация на устройство:

S'_S = {$Unregistered_S(ss_1)$, *ReadFwVersion*(ss_2), $Operational_S(ss_3)$, *DeliverFwUri*(ss_4), $Downloading_S(ss_5)$, *DownloadingStatus*(ss_6), *Updating*(ss_7), *StorageStatus*(ss_{12}), *Deregistering*(ss_9), *TransportBinding*(ss_{10}), *UpdateRegistration*(ss_{11}), *Disabling*(ss_{14}), *NotificationStoring*(ss_{13}), *Rebooting*(ss_8)};

Inp'_S = {$reg_{REQ}(ts_1)$, $Treg_S(ts_2)$, $deReg_{REQ}(ts_3)$, $Dereg_{RES}(ts_4)$, $readFw_{RES}$(*downloaded*)(ts_5), $readFw_{RES}$(*newAvailable*)(ts_6), *deregister*(ts_8), $readFw_{RES}$(*updatedFw*)(ts_7), $bind_{RES}(ts_9)$, *updateReg*(ts_{10}), $updateReg_{RES}(ts_{11})$, *fwAvailable*(ts_{12}), $updateFw_{RES}(ts_{17})$, $fwState_{RES}$(*downloaded*)(ts_{15}), $fwState_{RES}$(*downloading*) (ts_{16}), $reboot_{RES}(ts_{18})$, $disable_{ACK}(ts_{19})$, $disable_{RES}(ts_{20})$, $fwT_{DL}(ts_{14})$, $availMem_{RES}(ts_{21})$, $storing_{RES}(ts_{22})$, $Tdisable_S(ts_{23})$, $writeUri_{RES}(ts_{13})$};



Фиг. 2.2 Разширен модел на състоянието на регистрацията на устройство от гледна точка на самото устройство

$\rightarrow'_S$ = { τ_1^S, τ_2^S, τ_3^S, τ_4^S, τ_5^S, τ_6^S, τ_7^S, τ_8^S, τ_9^S, τ_{10}^S, τ_{11}^S, τ_{12}^S, τ_{13}^S, τ_{14}^S, τ_{15}^S, τ_{16}^S, τ_{17}^S, τ_{18}^S, τ_{19}^S, τ_{20}^S, τ_{21}^S, τ_{22}^S, τ_{23}^S, τ_{24}^S, τ_{25}^S };

s_S^0 = {$Unregistered_S$}, където

$\tau_1^S = (ss_1 ts_{23} ss_1)$; $\tau_2^S = (ss_1 ts_1 ss_2)$; $\tau_3^S = (ss_2 ts_3 ss_{17})$; $\tau_4^S = (ss_2 ts_6 ss_4)$;

$\tau_5^S = (ss_2 ts_5 ss_7)$; $\tau_6^S = (ss_3 ts_1 ss_3)$; $\tau_7^S = (ss_3 ts_2 ss_1)$; $\tau_8^S = (ss_3 ts_3 ss_1)$;
$\tau_9^S = (ss_3 ts_8 ss_1)$; $\tau_{10}^S = (ss_3 ts_{10} ss_{10})$; $\tau_{11}^S = (ss_3 ts_{12} ss_4)$; $\tau_{12}^S = (ss_3 ts_{19} ss_{12})$;
$\tau_{13}^S = (ss_9 ts_4 ss_1)$; $\tau_{14}^S = (ss_{10} ts_9 ss_{11})$; $\tau_{15}^S = (ss_{11} ts_{11} ss_1)$; $\tau_{16}^S = (ss_{12} ts_{21} ss_{13})$;
$\tau_{17}^S = (ss_{13} ts_{22} ss_{14})$; $\tau_{18}^S = (ss_{14} ts_{20} ss_1)$; $\tau_{19}^S = (ss_4 ts_{13} ss_6)$; $\tau_{20}^S = (ss_6 ts_8 ss_9)$;
$\tau_{21}^S = (ss_5 ts_{14} ss_6)$; $\tau_{22}^S = (ss_6 ts_{16} ss_5)$; $\tau_{23}^S = (ss_6 ts_{15} ss_7)$; $\tau_{24}^S = (ss_7 ts_{17} ss_8)$;
$\tau_{25}^S = (ss_8 ts_{18} ss_1)$.

Нека $R'_D = (S'_D, Inp'_D, \rightarrow'_S, s_D^0)$ бъде LTS, която изразява гледната точка на клиент относно състоянието на регистрация на устройство:

S'_D={$Unregistered_D(sd_1)$, $Deregistering_D(sd_6)$, $Operational_D(sd_3)$, $UpdateRegistration_D(sd_4)$, $Registering(sd_2)$, $FirmwareDownloading_D(sd_5)$} ;

Inp'_D = {$online(td_1)$, $T_{DISABLE}(td_2)$, $readFw_{REQ}(td_3)$, $sDereg_{REQ}(td_4)$, $reg_{RES}(td_5)$, $Treg_D(td_6)$, $updateReg_{REQ}(td_7)$, $softOffline(td_9)$, $bind_{REQ}(td_8)$, $deReg_{RES}(td_{10})$, $writeUri_{REQ}(td_{11})$, $readFw_{REQ}(td_{12})$, $updateFw_{REQ}(td_{13})$, $availMem_{REQ}(td_{14})$, $storing_{REQ}(td_{15})$, $disable_{REQ}(td_{16})$, $reboot_{REQ}(td_{17})$};

$\rightarrow'_D$ = { τ_1^D, τ_2^D, τ_3^D, τ_4^D, τ_5^D, τ_6^D, τ_7^D, τ_8^D, τ_9^D, τ_{10}^D, τ_{11}^D, τ_{12}^D, τ_{13}^D, τ_{14}^D, τ_{15}^D, τ_{16}^D, τ_{17}^D, τ_{18}^D, τ_{19}^D } ;

s_D^0 = {$Unregistered_D$}, където

$\tau_1^D = (sd_1 td_1 sd_2)$; $\tau_2^D = (sd_2 td_3 sd_2)$; $\tau_3^D = (sd_2 td_5 sd_3)$; $\tau_4^D = (sd_3 td_4 sd_1)$;
$\tau_5^D = (sd_3 td_6 sd_2)$; $\tau_6^D = (sd_3 td_8 sd_4)$; $\tau_7^D = (sd_3 td_9 sd_6)$; $\tau_8^D = (sd_3 td_{11} sd_5)$;
$\tau_9^D = (sd_5 td_7 sd_3)$; $\tau_{10}^D = (sd_3 td_{14} sd_3)$; $\tau_{11}^D = (sd_3 td_{15} sd_3)$; $\tau_{12}^D = (sd_3 td_{13} sd_3)$;
$\tau_{13}^D = (sd_3 td_{16} sd_1)$; $\tau_{14}^D = (sd_3 td_{17} sd_2)$; $\tau_{15}^D = (sd_5 td_3 sd_5)$; $\tau_{16}^D = (sd_6 td_{10} sd_1)$;
$\tau_{17}^D = (sd_4 td_7 sd_2)$; $\tau_{18}^D = (sd_1 td_2 sd_2)$; $\tau_{19}^D = (sd_5 td_4 sd_1)$.

Твърдение 2.2 Системите с именувани преходи R'_D и R'_S изразяват бисимулация помежду си.

Доказателство 2.2 Нека се създаде следната релация на съответните състояния:

$U_{R'S'}$ = { ($Unregistered_D$, $Unregistered_S$) , ($Operational_D$, $Operational_S$) },

Тогава за определени събития, свързани с управлението на регистрацията, могат да се идентифицират взаимовръзки между съответните преходи:

В случай на регистрация, когато устройството е включено и е с обновена версия на софтуера:

За $Unregistered_D$ { τ_1^D , τ_2^D , τ_3^D } $Operational_D$ $\exists$ $Unregistered_S$ { τ_1^S , τ_2^S , τ_3^S } $Operational_S$

В случай на регистрация, когато сървърът бива наличен:

За $Unregistered_D$ { τ_2^D , τ_3^D } $Operational_D$ $\exists$ $Unregistered_S$ { τ_1^S , τ_2^S , τ_3^S } $Operational_S$

В случай на инициране на де регистрация от страна на устройство:

За $Operational_D$ { τ_7^D , τ_{16}^D } $Unregistered_D$ $\exists$ $Operational_S$ { τ_8^S } $Unregistered_S$

В случай на инициране на дерегистрация от страна на сървър:

За $Operational_D$ { τ_4^D } $Unregistered_D$ $\exists$ $Operational_S$ { τ_9^S , τ_{13}^S } $Unregistered_S$

В случай на повторна регистрация поради изтичане на валидността на предходната регистрация:

За $Operational_D$ { τ_5^D , τ_3^D } $Operational_D$ $\exists$ $Operational_S$ { τ_6^S } $Operational_S$

В случай на обновяване на регистрацията:

За $Operational_D$ { τ_6^D , τ_{17}^D , τ_3^D } $Operational_D$ $\exists$ $Operational_S$ { τ_{10}^S , τ_{14}^S , τ_{15}^S , τ_2^S , τ_3^S } $Operational_S$

В случай на прекъсване на работата на сървъра:

За $Operational_D$ { τ_{10}^D , τ_{11}^D , τ_{13}^D } $Unregistered_D$ $\exists$ $Operational_S$ { τ_{12}^S , τ_{16}^S , τ_{17}^S , τ_{18}^S } $Unregistered_S$

В случай на обновяване на софтуерната версия:

За $Operational_D$ { τ_8^D , τ_{15}^D , τ_9^D , τ_{14}^D , τ_3^D } $Operational_D$ $\exists$ $Operational_S$ { τ_{11}^S , τ_{19}^S , τ_{21}^S , τ_{22}^S , τ_{23}^S , τ_{24}^S , τ_{25}^S , τ_2^S , τ_3^S} $Operational_S$

В случай на иницииране на дерегистрация по време на изтегляне на нов софтуерен пакет:

За $Operational_D$ { τ_8^D , τ_{15}^D , τ_{19}^D } $Unregistered_D$ $\exists$ $Operational_S$ { τ_{11}^S , τ_{19}^S , τ_{21}^S , τ_{22}^S , τ_{20}^S , τ_{13}^S } $Unregistered_S$

В случай на включено устройство, което използва стара софтуерна версия, но разполага с изтегления нов софтуерен пакет:

За $Unregistered_D$ { τ_1^D , τ_2^D , τ_3^D , τ_{12}^D } $Operational_D$ $\exists$$Unregistered_S$ { τ_2^S , τ_5^S , τ_{24}^S , τ_{25}^S, τ_2^S , τ_3^S } $Operational_S$

Изразените взаимовръзки между съответните преходи следователно осигуряват твърдението, че R'_D и R'_S са в състояние на бисимулация помежду си.■

Изследвани са функции за мениджмънт работоспособността в комуникации от машинен тип, които са описани в авторски публикации **[A4]** и **[A5].** Синтезирани и верифицирани са модели за мениджмънт на неизправности, мениджмънт на свързаността и управление на качеството на обслужване.

Възможностите на услуги за мениджмънт на устройства, които се предлагат от вървър на възможности на услуги (SCS), осигуряват общи функции на M2M приложенията, които могат да бъдат в мрежата (NA) или в M2M устройство (DA). В допълнение към функции за мениджмънт на неизправности, които могат да бъдат идентифицирани в различните приложни области, съществуват и общи такива, които могат да бъдат приложени към голяма група от M2M устройства. Тези функции могат да бъдат изразени посредством REpresentational State Transfer (REST) примитиви.

В контекста на мениджмънт на работоспособността съществуват три основни типа комуникационни модели.

Първият комуникационен модел е свързан със заявки за състоянието на ресурси или с актуализирането им. Вторият комуникационен модел е свързан с периодично докладване за работоспособността на едно M2M устройство. В този случай се изискват операции за управление на абонаментите и за периодичните известявания. Третият комуникационен модел е свързан с известявания при наличие на събития. Примери за операции, свързани с работоспособността, които могат да задействат известяване, са тези, които се отнасят към натоварването на M2M устройство. Политиката за управление на натоварването определя правилата, които NA трябва да следва за съответните устройства. Тя също така може да определи какви действия би следвало да се предприемат в случай на промяна в нивото на натоварване. Например за едно приложение, което функционира в реално време и е с критичен характер, поддържането на непрекъснатостта на услугата е от изключително значение. Нивото на натоварване на всяка цена би следвало да бъде под зададеното. В случай, че нивото на натоварване се повиши уместно за приложението бива прекъсването на връзката, като може да се направен повторен опит за свързване на по-късен етап.

В случай на активен абонамент M2M устройството би могло да извести NA при откриване на промяна в условията на натоварване (например при промяна на

нивото на натоварване от „0 към 1“, от „0 към 2“ или от „1 към 0“). В ниво на натоварването „0“, M2M устройството се справя успешно в рамките на собствените си спецификации за натоварване. В ниво на натоварването „1“ M2M устройството е претоварено. В ниво на натоварването „2“ M2M устройството е сериозно претоварено. Примери за операции, които биха могли да задействат известяване, са насочени към докладване на неизправности, свързани с едно M2M устройство. Фиг. 2.5 представя потока от съобщения за абонамент към известявания, известяване при настъпването на събитие и прекратяване на абонамента за известявания.

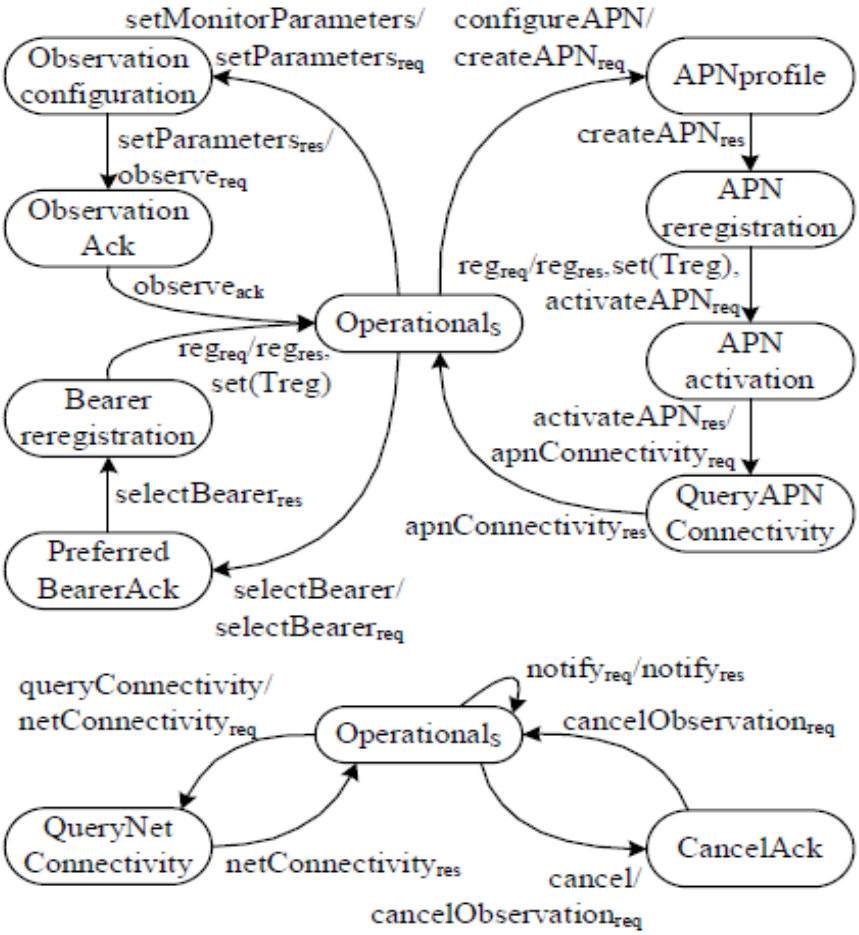
Табл.2.1 представя връзката между някои типове RESTful операции за уеб услуги и съответните HTTP (Hypertext Transfer Protocol) методи, които биха могли да бъдат използвани за осъществяването им.

Табл.2.1 Типове RESTful операции за уеб услуги и съответните HTTP методи

Операции на Web услуга	**HTTP метод**
Извличане на статуса на M2M ресурс	GET
Абонамент за събития(периодично докладване, докладване при наличие на събитие)	POST
Прекратяване на абонамент за събития	DELETE
Известяване при изменение на ресурс	PUT
Изпълнение на команда/задача за мениджмънт от ресурса	PUT

Изследвано е натоварването на SCS, което е резултат от трафик, свързан с мениджмънт на работоспособността и неизправностите на M2M устройства.

Едно устройство би могло да осъществи комуникация, използвайки следните клетъчни, безжични или жични носещи ресурси: Ethernet, DSL (Digital Subscriber Lines), PLC (Power Line Communications), Bluetooth, WLAN (Wireless Local Area Network), IEEE 802.15.4, WCDMA (Wideband Code Division Multiple Access), CDMA2000, WiMAX, GSM (Global System for Mobile communications), LTE и TD-SCDMA (Time Division Synchronous Code Division Multiple Access). Един сървър може да наблюдава изградената комуникационна линия и силата на сигнала от страна на устройството. За тази цел биват осъществени взаимоотношения за наблюдение помежду им, като се назначава съответната политика. Устройството изпраща периодично или по заявка съобщения с изискваната информация, докато изградените взаимоотношения не бъдат прекратени от страна на сървъра. Сървърът може да изиска информация за различни мрежови параметри, отнасящи се до свързаността на едно устройство, като например моментния използван мрежови ресурс или наличните други такива с цел оптимизиране на дадена свързаност. В случай, че едно устройство използва клетъчна мрежова свързаност, но е налична предпочитана и поддържана WLAN свързаност от даденото устройство, сървърът може да вземе решение за промяна на носещия ресурс към оптималния такъв. Сървърът също така може да създаде и да позволи използването на нов „профил на точка за достъп“ (Access Point Name – APN). Модел за управление на свързаността от гледна точка на сървър е представен на фиг. 2.7.



Фиг. 2.7 Модел за управление на свързаността от гледна точка на сървър

В състояние *Operational*$_S$ устройството е регистрирано и е в готовност за работа. В състояние *ObservationConfiguration* сървърът конфигурира политика за измервания, за които трябва да докладва устройството. В състояние *ObservationAck* сървърът изчаква за потвърждение, че наблюдението е активирано. В състояние *QueryNetConnectivity* сървърът изпраща запитване относно параметрите на свързаността и изчаква отговор. В състояние *PreferredBearerAck* сървърът заявява предпочитания носещ ресурс и изчаква за потвърждение. В състояние *BearerRegistration* сървърът изчаква регистриране от страна на устройството след избора на носещ ресурс. В състояние *APNprofile* сървърът изисква създаването на нов APN профил и изчаква за потвърждение. В състояние *APNregistration* сървърът изчаква повторната регистрация на устройството след избора на новия APN профил. В състояние *APNactivation* сървърът активира новата APN и изчаква за потвърждение. В състояние *QueryAPNConnectivity* сървърът изчаква заявената информация относно APN свързаността. В състояние *CancelAck* сървърът прекратява наблюдението и изчаква за потвърждение.

Нека $CM_S = (S_S, Inp_S, \rightarrow_S, s_S^0)$ е LTS, която представя гледната точка на сървър за модела на управление на свързаността, а именно:

S_S = {*Operational*$_S$, *ObservationConfiguration*, *ObservationAck*, *QueryNetConnectivity*, *PreferredBearerAck*, *BearerRegistration*, *CancelAck*, *APNregistration*, *APNprofile*, *APNactivation*, *QueryAPNConnectivity*};

Inp_S = {*queryConnectivity*, $netConnectivity_{RES}$, *setMonitorParameters*, $notify_{REQ}$, $setParametes_{RES}$, $observe_{ACK}$, *selectBearer*, $selectBearer_{RES}$, reg_{REQ}, *configureAPN*, $createAPN_{RES}$, $activateAPN_{RES}$, $apnConnectivity_{RES}$, *cancel*, $cancelObservation_{RES}$};

$\rightarrow_S$ = { τ_1^S, τ_2^S, τ_3^S, τ_4^S, τ_5^S, τ_6^S, τ_7^S, τ_8^S, τ_9^S, τ_{10}^S, τ_{11}^S, τ_{12}^S, τ_{13}^S, τ_{14}^S, τ_{15}^S, τ_{16}^S} ;

s_S^0 = {$Operational_S$}, където

τ_1^S = (*Operational*$_S$ *setMonitorParameters ObservationConfiguration*);

τ_2^S = (*ObservationConfiguration* $setParametes_{RES}$ *ObservationAck*);

τ_3^S = (*ObservationAck* $observe_{ACK}$ *Operational*$_S$);

τ_4^S = (*Operational*$_S$ $notify_{REQ}$ *Operational*$_S$);

τ_5^S = (*Operational*$_S$ *queryConnectivity QueryNetConnectivity*);

τ_6^S = (*QueryNetConnectivity* $netConnectivity_{RES}$ *Operational*$_S$);

τ_7^S = (*Operational*$_S$ *selectBearer PreferredBearerAck*);

τ_8^S = (*PreferredBearerAck* $selectBearer_{RES}$ *BearerRegistration*);

τ_9^S = (*BearerRegistration* reg_{REQ} *Operational*$_S$);

τ_{10}^S = (*Operational*$_S$ *configureAPN APNprofile*);

τ_{11}^S = (*APNprofile* $createAPN_{RES}$ *APNregistration*);

τ_{12}^S = (*APNregistration* reg_{REQ} *APNactivation*);

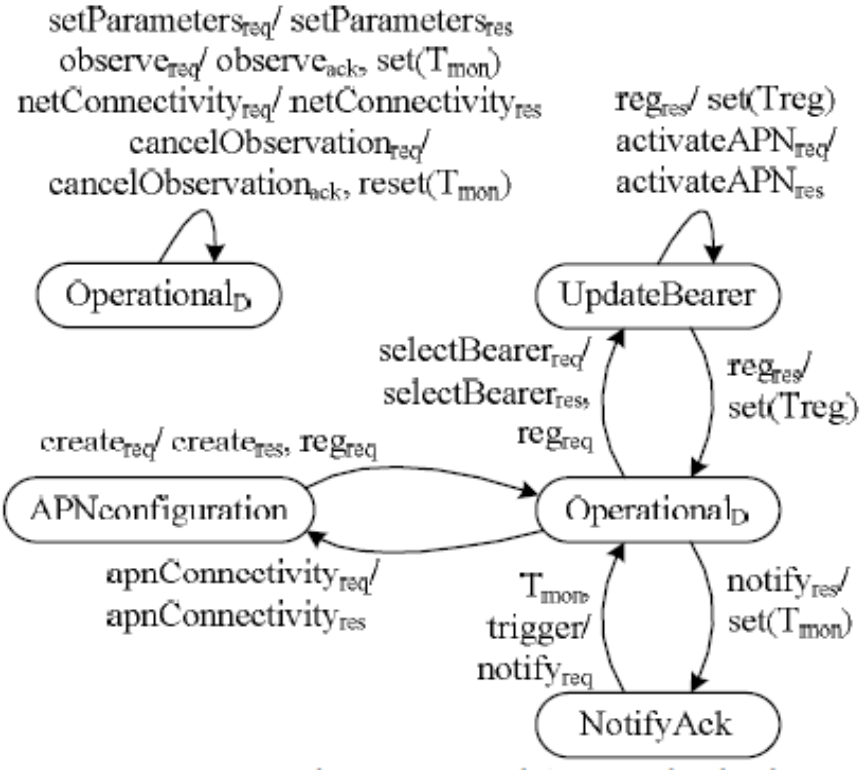
τ_{13}^S = (*APNactivation* $activateAPN_{RES}$ *QueryAPNConnectivity*);

τ_{14}^S = (*APNactivation* $apnConnectivity_{RES}$ *Operational*$_S$);

τ_{15}^S = (*Operational*$_S$ *cancel CancelAck*);

τ_{16}^S = (*CancelAck* $cancelObservation_{RES}$ *Operational*$_S$);

Модел за управление на свързаността от гледна точка устройство е представен на фиг. 2.8.



Фиг. 2.8 Модел за управление на свързаността от гледна точка на устройство

В състояние $Operational_S$ устройството е регистрирано и е в готовност за работа. В това състояние сървърът може да наложи политика за наблюдение, да я активира и последствие да прекъсне наблюдението. В състояние $Operational_D$

устройството също така може да изпраща отговори на заявки, свързани с мрежовата свързаност. В състояние *NotifyAck* устройството информира сървъра за изисканата информация и изчаква отговор. В състояние *UpdateBearer* устройството е в процес на смяна на използвания носещ реусрс и последваща повторна регистрация. В състояние *APNconfiguration* устройството е в процес на създаване и установяване на нов APN профил.

Нека $CM_D = (S_D, Inp_D, \rightarrow_D, s_D^0)$ е LTS, която представя гледната точка на устройство за модела на мениджмънт на свързаността му, а именно:

S_D = {*Operational*$_D$, *NotifyAck*, *UpdateBearer*, *APNconfiguration*};

Inp_D = {*setParameters*$_{REQ}$, *observe*$_{REQ}$, *netConnectivity*$_{REQ}$, *cancelObservation*$_{REQ}$, T_{MON}, *trigger*, *notify*$_{RES}$, *selectBearer*$_{REQ}$, *reg*$_{RES}$, *activateAPN*$_{REQ}$, *create*$_{REQ}$, *apnConnectivity*$_{RES}$};

$\rightarrow_D$ = { τ_1^D, τ_2^D, τ_3^D, τ_4^D, τ_5^D, τ_6^D, τ_7^D, τ_8^D, τ_9^D, τ_{10}^D, τ_{11}^D, τ_{12}^D, τ_{13}^D};

s_D^0 = {*Operational*$_D$}, където

τ_1^D = (*Operational*$_D$ *setParameters*$_{REQ}$ *Operational*$_D$);

τ_2^D = (*Operational*$_D$ *observe*$_{REQ}$ *Operational*$_D$);

τ_3^D = (*Operational*$_D$ *netConnectivity*$_{REQ}$ *Operational*$_D$);

τ_4^D = (*Operational*$_D$ *cancelObservation*$_{REQ}$ *Operational*$_D$);

τ_5^D = (*Operational*$_D$ *selectBearer*$_{REQ}$ *UpdateBearer*);

τ_6^D = (*UpdateBearer reg*$_{RES}$ *Operational*$_D$);

τ_7^D = (*Operational*$_D$ T_{MON} *NotifyAck*);

τ_8^D = (*Operational*$_D$ *trigger NotifyAck*);

τ_9^D = (*NotifyAck notify*$_{RES}$ *Operational*$_D$);

τ_{10}^D = (*Operational*$_D$ *create*$_{REQ}$ *APNconfiguration*);

τ_{11}^D = (*APNconfiguration reg*$_{RES}$ *APNconfiguration*);

τ_{12}^D = (*APNconfiguration activateAPN*$_{REQ}$ *APNconfiguration*);

τ_{13}^D = (*APNconfiguration apnConnectivity*$_{RES}$ *Operational*$_D$);

Твърдение 2.3 Системите с именувани преходи CM_S и CM_D изразяват бисимулация помежду си.

Доказателство 2.3 По дефиниция, за да се докаже, че двете системи изразяват бисимулация помежду си, трябва да се идентифицира съответствие между техните състояния. Нека се създаде следната релация на съответните състояния:
U_{DS} = {(*Operational*$_D$, *Operational*$_S$), (*UpdateBearer*, *PreferredBearerAck*), (*APNconfiguration*, *APNprofile*)}, тогава:

В случай на задаване на политика за наблюдение: За *Operational*$_D$ $\exists$ { τ_1^D , τ_2^D } *и за Operational*$_S$ $\exists$ { τ_1^S , τ_2^S , τ_3^S }

В случай на периодично известяване относно състоянието на сигнала и изградената линия: За *Operational*$_D$ $\exists$ { τ_7^D , τ_9^D } *и за Operational*$_S$ $\exists$ { τ_4^S }

В случай на изискано известяване относно състоянието на сигнала и изградената линия:
За *Operational*$_D$ $\exists$ { τ_8^D , τ_9^D } *и за Operational*$_S$ $\exists$ { τ_4^S }

В случай на прекратяване на наблюдението: За *Operational*$_D$ $\exists$ { τ_4^D} *и за Operational*$_S$ $\exists$ { τ_{15}^S , τ_{16}^S }

В случай на изискване на информация за мрежовата свързаност: За *Operational*$_D$ $\exists$ { τ_3^D} *и за Operational*$_S$ $\exists$ { τ_5^S , τ_6^S }

В случай на избор на предпочитан носещ ресурс: За $Operational_D \exists \{\tau_5^D\}$ *и за* $Operational_S \exists \{\tau_7^S\}$

В случай на повторна регистрация след избор на най-добрия носещ ресурс: За *UpdateBearer* $\exists \{\tau_6^D\}$ *и за PreferredBearerAck* $\exists \{\tau_8^S, \tau_9^S\}$

В случай на създаване на нов APN профил: За $Operational_D \exists \{\tau_{10}^D\}$ *и за* $Operational_S \exists \{\tau_{10}^S\}$

В случай на повторна регистрация, активиране на APN и проверка на APN свързаността: За *APNconfiguration* $\exists \{\tau_{11}^D, \tau_{12}^D, \tau_{13}^D\}$ *и за APNprofile* $\exists \{\tau_{11}^S, \tau_{12}^S, \tau_{13}^S, \tau_{14}^S\}$

От изразените взаимовръзки следва, че системи с именувани преходи CM_S и CM_D изразяват бисимулация помежду си, което от своя страна води до това, че моделите, свързани с управлението на свързаността са синхронизирани.■

В авторска публикация **[A6]** е синтезиран модел на автономен агент, който управлява избора на носещ ресурс. Агентът взаимодейства с M2M устройствата, възприемайки информация от тях, като паралелно с това устройствата преминава през поредица от състояния, с цел поддържане на максимални стойности на производителността. Самооптимизирането е процес на дефиниране на цел с използване на набор от средства за изпълнението й. Една от основните цели на самооптимизирането в комуникации от машинен тип е избор на най-добрия носещ ресурс, който да бъде използван. Агент, който е предназначен за M2M самооптимизиране, е в състояние да разсъди върху определени действия и да предприеме оптималните, за да постигне предефинираната цел.

Използвана е темпорална логика за описание на поведението на автономен агент, който отговаря за мениджмънт на свързаността. След регистриране на M2M устройство агентът конфигурира различни по тип филтри, които отразяват политиката за избор на носещ ресурс. Поведението на автономния агент за мениджмънт на свързаността е описано с уравнения от (2.22) до (2.52):

$$G(\text{Unregistered}(x) \rightarrow \top U \text{reg}_{\text{REQ}}(x)) \quad (2.22)$$

$$G(\text{Unregistered}(x) \wedge \text{reg}_{\text{REQ}}(x) \rightarrow \text{reg}_{\text{RES}}(x) \wedge \neg \text{BadPreferred}(x) \wedge \text{configGeoTrap}_{\text{RES}}(x) \wedge N\text{WaitGeoAck}(x)) \quad (2.23)$$

$$G(\text{WaitGeoAck}(x) \rightarrow \top U \text{configGeoTrap}_{\text{RES}}(x)) \quad (2.24)$$

$$G(\text{WaitGeoAck}(x) \wedge \text{configGeoTrap}_{\text{RES}}(x) \rightarrow \text{configPowerTrap}_{\text{REQ}}(x) \wedge N\text{WaitPowerAck}(x)) \quad (2.25)$$

$$G(\text{WaitPowerAck}(x) \rightarrow \top U \text{configPowerTrap}_{\text{RES}}(x)) \quad (2.26)$$

$$G(\text{WaitPowerAck}(x) \wedge \text{configPowerTrap}_{\text{RES}}(x) \rightarrow \text{getLocation}_{\text{REQ}}(x) \wedge N\text{WaitLocation(x)}) \quad (2.27)$$

$$G(\text{WaitLocation}(x) \rightarrow \top U \text{getLocation}_{\text{RES}}(x)) \quad (2.28)$$

$$G(\text{WaitLocation}(x) \wedge \text{getLocation}_{\text{RES}}(x) \rightarrow \text{connParameters}_{\text{REQ}}(x) \wedge N\text{WaitConnectivity}(x)) \quad (2.29)$$

$$G(\text{WaitConnectivity}(x) \rightarrow \top U \text{connParameters}_{\text{RES}}(x)) \quad (2.30)$$

$$G(\text{WaitConnectivity}(x) \wedge \text{connParameters}_{\text{RES}}(x) \wedge \text{InArea}(a, x) \wedge \text{PreferredIn}(b, a) \wedge \text{Used}(b, x) \wedge (\text{Excellent}(b, x) \vee \text{Good}(b, x)) \rightarrow N\text{Operational}(x)) \quad (2.31)$$

$$G(\text{WaitConnectivity}(x) \wedge \text{connParameters}_{\text{rRES}}(x) \wedge \text{InArea}(a,x) \wedge \text{PreferredIn}(b,a) \wedge \text{Used}(b,x) \wedge \text{Bad}(b,x) \wedge \text{Available}(c,x) \wedge \text{Best}(c,a) \rightarrow \text{select}_{\text{REQ}}(x,c) \wedge \text{BadPreferred}(x) \wedge N\text{WaitBearerAck}(x)) \quad (2.32)$$

$$G(\text{WaitConnectivity}(x) \wedge \text{connParameters}_{\text{RES}}(x) \wedge \text{InArea}(a,x) \wedge \text{Used}(b,x) \wedge$$

$\neg$PreferredIn(b,a)$\wedge$Available(c,x)$\wedge$PreferredIn(c,a)$\wedge\neg$BadPreferred(x)$\rightarrow$
select$_{REQ}$(x,c)$\wedge N$WaitBearerAck(x)) (2.33)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge$InArea(a,x)$\wedge$
Used(b,x)$\wedge\neg$PreferredIn(b,a)$\wedge$Available(c,x) $\wedge$PreferredIn(c,a)$\wedge$BadPreferred(x)$\wedge$
Best(d,a)$\wedge$ Available(d,x) $\rightarrow$ select$_{REQ}$(x,d)$\wedge N$WaitBearerAck(x)) (2.34)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge$ $\neg$InArea(a, x)$\wedge$
PreferredOut(b,a)$\wedge$Used(b,x)$\wedge$(Excellent(b, x)$\vee$Good(b,x))$\rightarrow N$Operational(x)) (2.35)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge$ $\neg$InArea(a,x)$\wedge$
PreferredOut(b,a)$\wedge$Used(b,x)$\wedge$Bad(b,x)$\wedge$Available(c,x)$\wedge$Best(c,a)$\rightarrow$select$_{REQ}$(x,c)$\wedge$
BadPreferred(x)$\wedge N$WaitBearerAck(x)) (2.36)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge\neg$InArea(a,x)$\wedge$Used(b,x)$\wedge$
$\neg$PreferredOut(b,a)$\wedge$ Available(c, x)$\wedge$PreferredIn(c, a)$\wedge\neg$BadPreferred(x) $\rightarrow$
select$_{REQ}$(x,c)$\wedge$ NWaitBearerAck(x)) (2.37)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x) $\wedge$ $\neg$InArea(a,x)$\wedge$Used(b,x)$\wedge$
$\neg$PreferredOut(b,a)$\wedge$Available(c,x)$\wedge$PreferredIn(c,a)$\wedge$BadPreferred(x)$\wedge$Best(d,a)$\wedge$
Available(d,x) $\rightarrow$select$_{REQ}$(x, d)$\wedge N$WaitBearerAck(x)) (2.38)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge$Used(b,x) $\wedge$(Excellent(b, x)$\vee$
Good(b, x))$\wedge$AvailableEmpty(x)$\rightarrow$ NOperational(x)) (2.39)

G(WaitConnectivity(x)$\wedge$connParameters$_{RES}$(x)$\wedge$Used(b,x)$\wedge$
Bad(b,x)$\wedge$AvailableEmpty(x)$\rightarrow N$Unregistered(x)) (2.40)

G(WaitBearerAck(x)$\rightarrow \top U$selectBearer$_{REQ}$(x, b)) (2.41)

G(WaitBearerAck(x)$\wedge$selectBearers$_{RES}$(x)$\rightarrow N$WaitReregistration(x)) (2.42)

G(WaitReregistration(x)$\rightarrow \top U$reg$_{REQ}$(x)) (2.43)

G(WaitReregistration(x)$\wedge$reg$_{REQ}$(x)$\rightarrow$reg$_{RES}$(x)$\wedge$ $\neg$BadPreferred(x)$\wedge$
GetLocation$_{REQ}$(x)$\wedge N$WaitLocation(x)) (2.44)

G(Operational(x)$\rightarrow \top U$(notifyPower$_{REQ}$(x)$\vee$notifyGeo$_{REQ}$(x))) (2.45)

G(Operational(x)$\wedge$notifyGeo$_{REQ}$(x)$\rightarrow$notifyGeo$_{REQ}$(x)$\wedge$connParameters$_{REQ}$(x)$\wedge$
NWaitConnectivity(x)) (2.46)

G(Operational(x)$\wedge$notifyPower$_{REQ}$(x,b)$\wedge$PowerTrapInactive(x)$\rightarrow$
notifyPower$_{RES}$(x,b)$\wedge N$Operational(x)) (2.47)

G(Operational(x)$\wedge$notifyPower$_{REQ}$(x,b)$\wedge$PowerTrapActive(x)$\rightarrow$
notifyPower$_{RES}$(x,b)$\wedge$set(Tmargin(x))$\wedge N$WaitMargin(x)) (2.48)

G(WaitMargin(x)$\rightarrow \top U$(notifyPower$_{REQ}$(x,b)$\vee$notifyGeo$_{REQ}$(x)$\vee$Tmargin(x))) (2.49)

G(WaitMargin(x)$\wedge$notifyPower$_{REQ}$(x,b)$\wedge$PowerTrapInactive(x)$\rightarrow$
notifyPower$_{RES}$(x,b)$\wedge$resetTmargin(x)$\wedge N$Operational(x)) (2.50)

G(WaitMargin(x)$\wedge$Tmargin(x) $\rightarrow$connParameters$_{REQ}$(x)$\wedge N$WaitConnectivity(x)) (2.51)

G(WaitMargin(x)$\wedge$notifyGeo$_{REQ}$(x)$\rightarrow$notifyGeo$_{RES}$(x)$\wedge N$WaitMargin(x)) (2.52)

Идентифицирани са изискванията към управление на качеството на обслужване в комуникации от машинен тип и е създаден модел на автономен агент за управление на качеството на обслужване, представен в авторска публикация **[A7**].

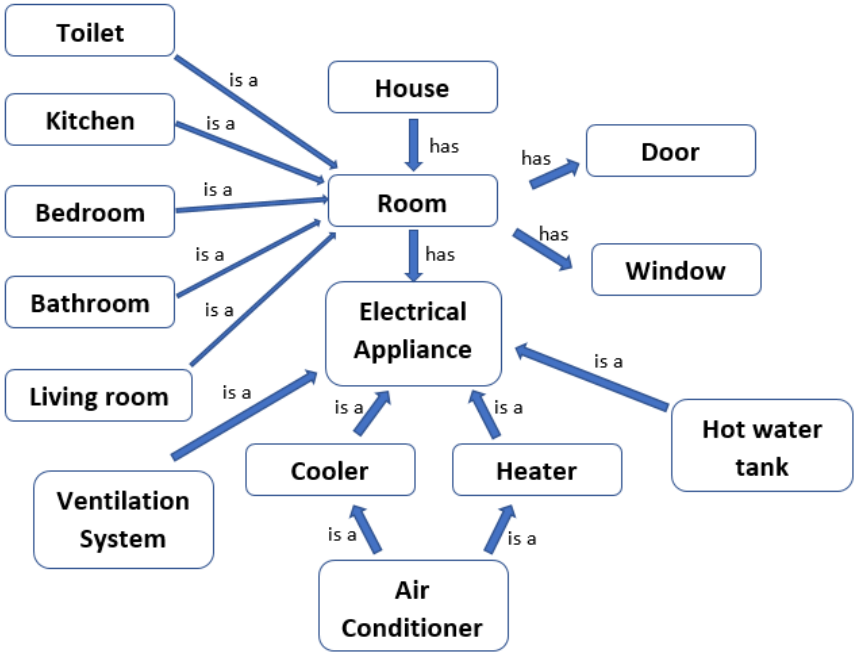
Резултатите от изследванията може да се използват в системи за мениджмънт на M2M устройства, като по този начин улесняват процесите при поддържане на жизнения цикъл на системата. Предимствата на използването на предложените модели в сравнение със съществуващите, специфични такива са, че са базирани на

широко приета рамка за мениджмънт, която е технологично независима и процедурите по мениджмънт могат да бъдат автоматизирани.

Трета глава. Изследване на автономни енергийни системи

Моделите на данни, които могат да бъдат използвани от различни по вид приложения, биват ключов фактор за осъществяването на множеството възможности, предоставени от услугите в комуникациите от машинен тип. Изследванията в трета глава са свързани с едно от най-важните приложения в света на „Интернет на нещата“ (Internet of Things-IoT), а именно интелигентното отчитане на електроенергия. Целта се корени в ефикасното управление на електроенергийните източници посредством моделиране на уеб услуги с техните ресурси и състояния.

Синтезиран е базиран на онтология модел за управление на електрически устройства в едно домакинство (фиг.3.2). Моделът представя познанието за областта под формата на общи идеи и взаимовръзки между тях, които биват описани в авторска публикация **[A8]**.

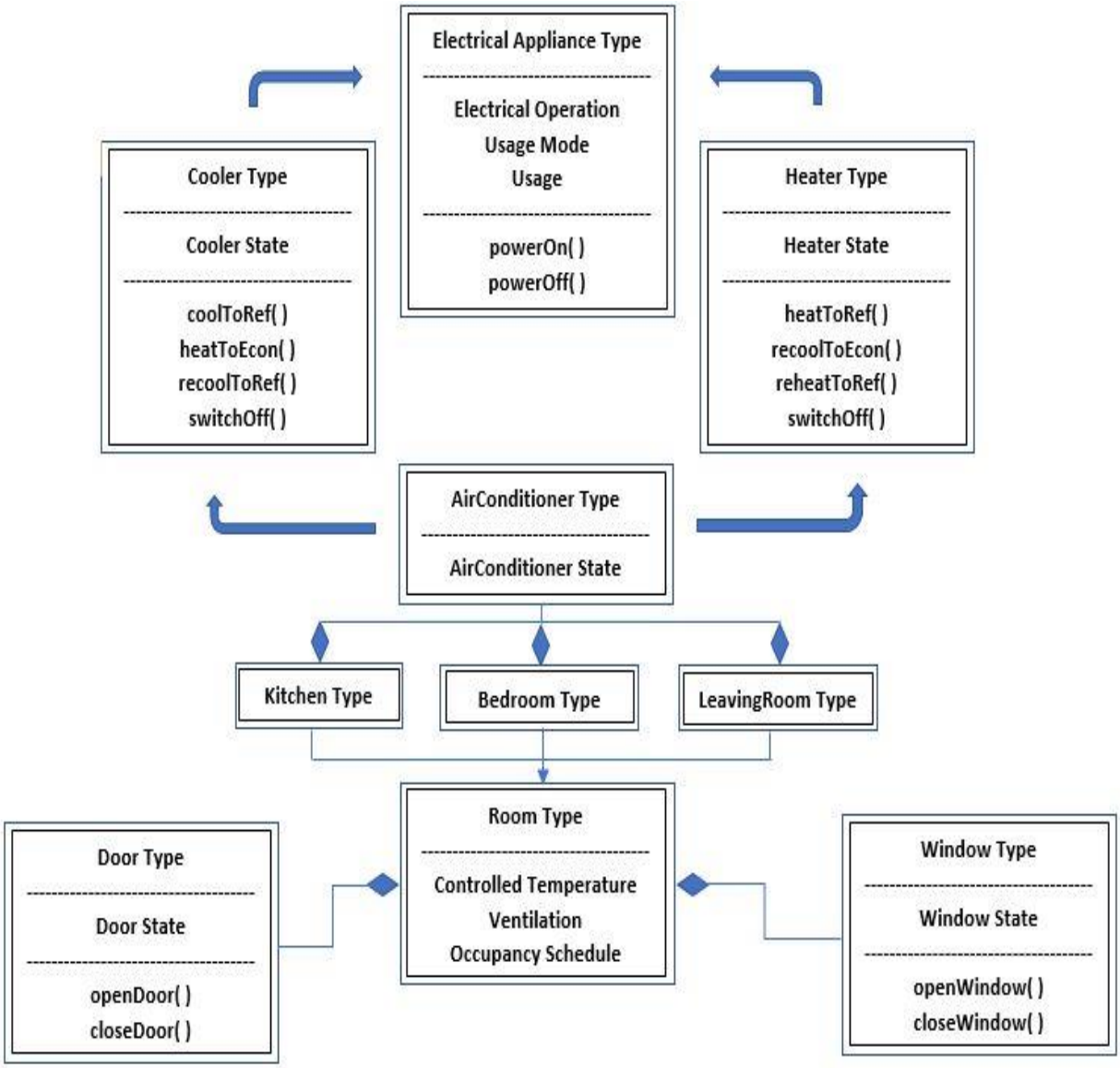


Фиг. 3.2 Онтология за управление на мощността в една интелигентна среда

Моделът дефинира типове *„неща“* с техните атрибути и операции (фиг.3.3)

. За създаването и проверката на онтологията към модела за управление на мощността в една интелигентна среда е използван софтуерният продукт Protégé. Protégé е в пълно съответствие със стандартите за създаване на OWL 2.

Предложен е метод за моделиране на семантична информация с цел спестяване на електроенергия, който е представен в авторска публикация **[A9].** Изложеният информационен модел за дадената област се фокусира не само върху сензорите и задвижващите устройства от областта, но и върху възможността за повсеместното им интегриране в една система от света на IoT.



Фиг. 3.3 Модел на информация за управляеми електрически устройства

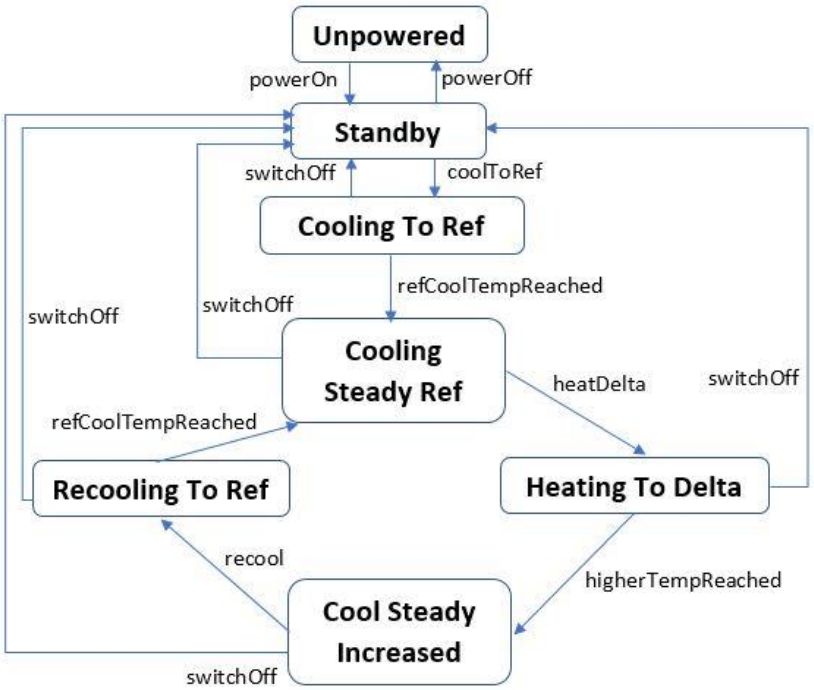
Фиг.3.6представя фрагмент от OWL описанието на модела.

```
Declaration ( Datatype ( :UsageMetadata )
Declaration ( Datatype ( :ElectricalOperationMetadata )
Declaration ( Datatype ( :UsageModeMetadata )
DatatypeDefinition (
        :UsageMetadata
        DataOneOf ( „1“^^xsd:interger „2“^^xsd:integer „3“xsd:integer
        Annotation (rdfs:comment „1 means permanent, 2 means semi-random, 3 means periodic“ )
        )
)
DataPropertyRande ( :requiresDelta xsd:nonNegativeFloat)
DataPropertyRande ( :hasAirConditionerID xsd:nonNegativeInteger)
```

Фиг. 3.6 OWL деклариране на типове данни и дефиниции към информационния модел на една климатична инсталация

Синтезирани са семантични данни за интелигентно измерване и отчитане на потреблението на електроенергия. При моделирането на семантична информация, свързана със спестяването на електроенергия, и разглеждайки изискванията на един потребител, се наблюдава нужда от функционалност за самоконфигуриране. Все пак потребителят би следвало да остане в състояние да добавя и конфигурира оборудване и услуги, да управлява и оптимизира мрежата в жилищната си среда. Процесът, който се нуждае от оптимизация посредством внедряването на автоматизация и минимизирането на човешкия фактор, включва конфигурирането на мрежовото оборудване на база потребителското оборудване, абонаментните профили, предпочитанията и потребителските услуги за достъпа до мрежата. Функционалността за самоконфигуриране е ключово изискване при моделирането на семантична информация и за доставчика на мрежата за достъп. За да се оптимизира реализирането на нови услуги посредством онлайн абонаменти, се изисква включването на автоматизиране в процесите на предоставяне и конфигуриране на потребителското оборудване. Изискванията от страна на производителите при моделиране на семантична информация са свързани с реализирането на енергийно ефективни комуникации, чиито механизъм да може бъде автоматизиран, като се цели ограничаване на броя на обменените съобщения.

Фиг. 3.11 илюстрира модел на състоянията на климатична инсталация, използвана за охлаждане, която може да приеме състояние *Powered* или *Unpowered*. В състояние *Unpowered* климатичната инсталация не е оперативна и съответно не може да бъде контролирана. За да премине в състояние на оперативност *Powered* е нужна човешка намеса, което от своя страна е разделено на две под състояния. В състояние *Standby* климатичната инсталация е оперативна, но все още не се използва за охлаждане, а е в процес на изчакване на инструкции. Охладителното тяло преминава в състояние *On*, когато получи инструкции да охлади до предпочитаната температура. В случай че бъде приложен механизъм за управление на мощността, състоянието *On* също бива разделено на две под състояния. Посредством операцията *coolToRef* се осъществява преминаването от състояние *Standby* към състояние *CoolingToRef*, като целта е климатичната инсталация да достигне зададената еталонна температура T_{ref}, консумирайки енергия. В момента на достигане на еталонната температура устройството уведомява за събитието, извиквайки операцията *refCoolTempReached* и преминава в състояние *CoolSteadyRef*, като в това състояние се консумира енергия за поддържане на еталонната температура. В определен интервал от денонощието се реализира идеята за минимална корекция на еталонната температура в предварително дефинирани допустими стойности, което се изразява в преминаване посредством операцията *heatDelta* в състояние на климатичната инсталация *HeatingToDelta*, с цел консумацията на енергия да бъде преустановена и да бъде достигната икономична температура, за която се известява чрез операцията *higherTempReached* и се преминава в състояние *CoolSteadyIncreased*. За разлика от състоянието *CoolSteadyRef*, в състояние *CoolSteadyIncreased* се използва по-малко енергия. Когато интервалът от денонощието, свързан с ограничава на консумацията на електроенергия приключи, отново се цели достигане и поддържане на еталонната температура чрез операциите *recool* и *refCoolTempReached* и преминаването през съответните състояния *RecoolingToRef* и *CoolSteadyRef*.



Фиг. 3.11 Модел на състоянията на охладително тяло

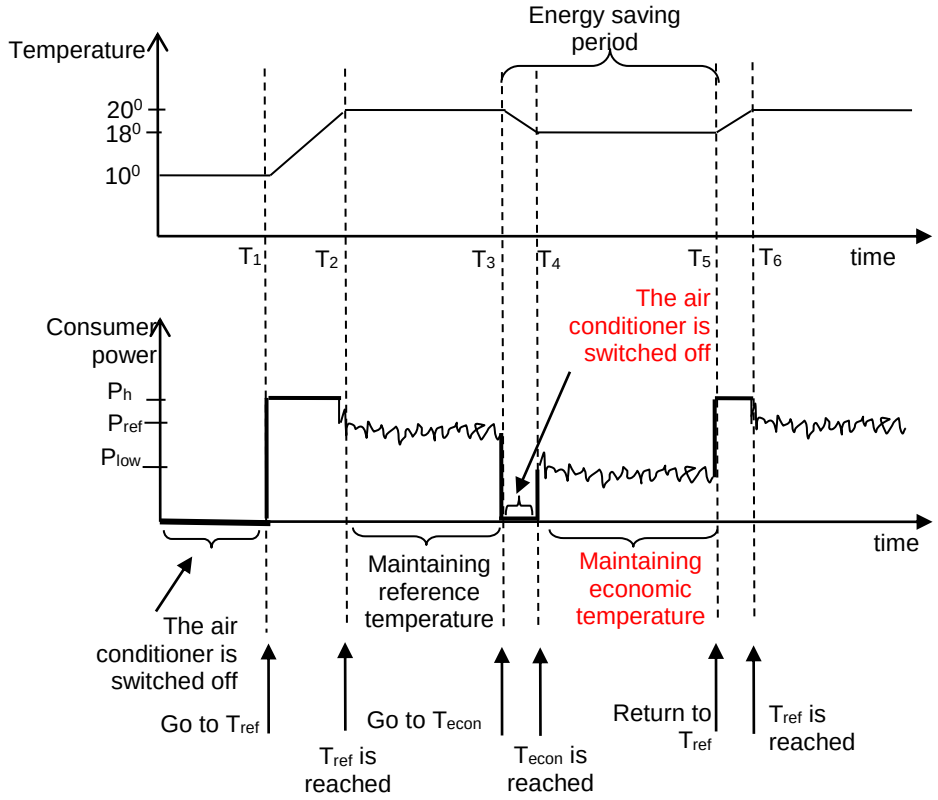
В авторска публикация **[A10]** е представен метод за дефиниране на приложни програмни интерфейси за управление на енергийната мощност на електрически уреди.

Дефинирана е уеб услуга Power Control Service (PCS), която може да се използва за управлението на мощността на климатична инсталация за охлаждане или затопляне. Уеб услугата PCS предоставя интерфейсите CoolingControl, SubscriptionManagement и TemperatureNotification, всеки от които поддържа определени операции. Интерфейсът CoolingControl се поддържа от управляеми електрически устройства, които притежават система за охлаждане. Интерфейсът SubscriptionManagement се използва за управление на абонаментите за известия, свързани с промяната в температурата. Интерфейсът TemperatureNotification осигурява операции за известяване на приложенията за настъпване на температурни събития.

Принципът на охлаждане е илюстриран на фиг. 3.15.

Нека се приеме, че охлаждащото тяло е в оперативно състояние, но неговата охлаждаща система е изключена, а температурата в помещението, където се помещава устройството е 30°C. Когато охлаждащата система бива задействана (T_1), то тя получава инструкции да охлади помещението до предпочитаната температура T_{ref} (20°C). Охлаждащата система консумира енергия P_c с цел да се достигне T_{ref}. Когато T_{ref} бива достигната (T_2), охлаждащата система изразходва енергия P_{ref}, поддържайки предпочитаната температура. Следвайки предварително изготвен график, се прилага управление на охлаждането, като при настъпването на T_3, в който предпочитаната температура спада, бива получена инструкция, с която охлаждащата система бива изключена и се предполага затопляне на помещението в рамките на минимална стойност Δ и достигане на $T_{ref+\Delta}$. През периода на затопляне T_3-T_4 не се консумира енергия. Когато се достигне $T_{ref+\Delta}$ в T_4, охлаждащата система се задейства, целейки поддържане на по-високата температура $T_{ref+\Delta}$. По време на

периода на пестене на енергия T_5-T_4 охлаждащата система консумира ограничено количество енергия P_{low}. В края на периода за пестене на електроенергия T_5 охлаждащата система получава команда да охлади до предпочитаната температура, като отново бива консумирана енергия P_c, чиято цел е достигането на T_{ref} и по този начин цикълът бива затворен.



Фиг. 3.15 Механизъм за управление на охлаждането

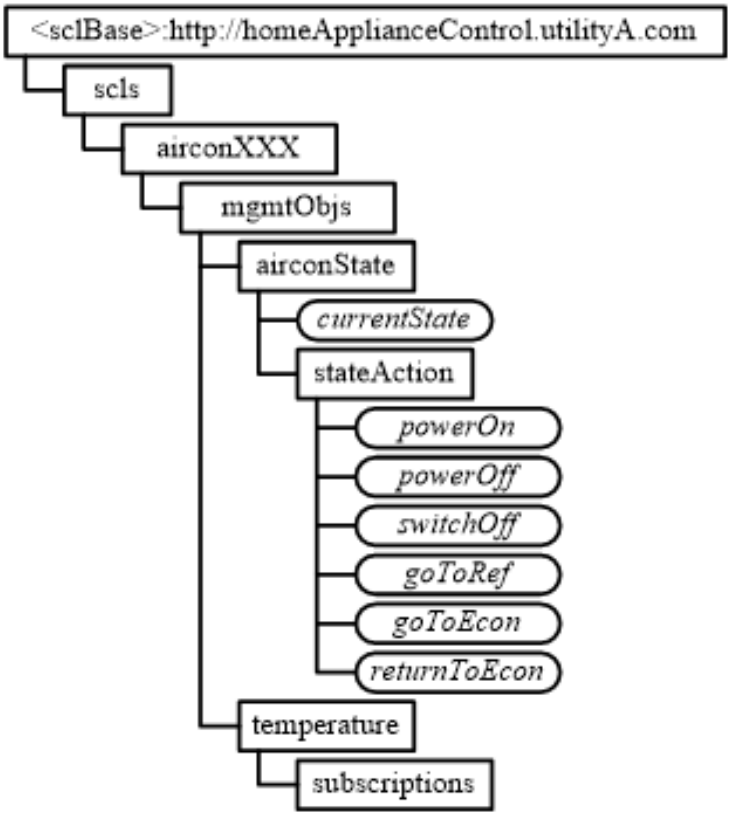
За устройства с ограничени ресурси като изчислителна мощ и памет се използва архитектурният стил REST. Възможностите за отдалечено управление на едно електрическо устройство например климатична инсталация в дома би следвало да бъдат разгърнати едновременно в мрежата и на самото устройство, реализирайки съответно свойствата на възможности на услуги **homeApplianceControl в** мрежата и **airconXXX** в устройството. Подобна функционалност позволява конфигурирането на предефиниран график върху управлението на мощността, който да рефлектира в ефикасно управление на консумацията на електроенергия.

Фиг. 3.19 представя структурата на ресурсите, отразяващи състоянията на една климатична инсталация.

Проектирани са две уеб услуги за управление на електрическите устройства в дома. *Schedule Management* е уеб услуга, която осигурява интерфейси за отдалечено конфигуриране на графика за управление на охлаждането и затоплянето.

Интерфейсът *season-SchedulesMgmt* позволява на NAs да получават и обработват информация, свързана с ресурса *SeasonSchedules*. Приемането и обработването на информация, свързана с *dailySchedules* се осъществява посредством интерейса *dailyScheduleMgmt*. Интерфейсът *dailyPeriodsMgmt* позволява обработката на информация, свързана с ресурса *dailyPeriods*.

Уеб услугата *Appliance Power Control* предоставя интерфейси, които биват обобщени в табл. 3.4, за манипулиране на състоянията на климатичната инсталация.



Фиг. 3.19 Структура на ресурси на състоянията на една климатична инсталация

Оценката на потенциала за управление на мощността, постигната посредством използването на уеб услуги, е базирана на методологията, свързана с термалните характеристики в едно домакинство. Сравнени са два отделни случая:

- зададен е график за два пикови температурни (21°) интервала, а именно между 06:30-09:00 и 17:30-22:00, докато препоръчаната температура би следвало да бъде поддържана по-ниска (20°) в останалата част от деня;
- поддържане на постоянна препоръчана температура 22°.

Предполага се, че стаята е с размер 30 м^3, прозорците са с големина 4,6 м^2, оборудвани с 5kW устройства. Пропускателната способност на стените и прозорците е съответно около 1 и 5 W/m2K. Охлаждането следва закона на Нютон и топлинната термална маса на помещението C_T е 107 J/K.

Методът за дефиниране на уеб услуги за енергийна ефективност и оценката от прилагането им са представени в авторски публикации **[A11]** и **[A12].**

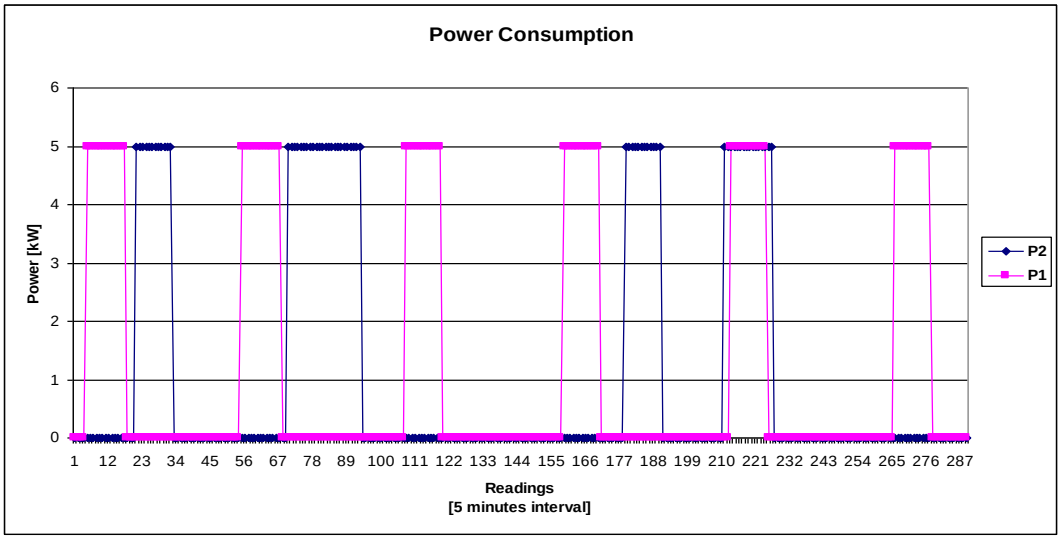
На фиг. 3.22 е илюстрирана консумацията на мощност спрямо времето, когато е приложено традиционно управление и базирано на IoT график. Може да се отбележи, че дискомфортът на съдържателите на жилището при поддържането на два градуса по-ниска температура посредством въвеждането на базиран на уеб услуги график, води до намаляване на консумацията на електроенергия.

За да се демонстрира експериментът в по-голям мащаб, нека се вземе предвид град с общо 600000 домакинства, където 5% от домакинствата използват климатични инсталации и 20% от тях биха приели използването на уеб услуги. Демонстрирайки разликата само при едно домакинство може да се предположи, че разликата ще окаже ефект на дневна и седмична база в рамките на съответно 25 и

125 MWh. За сравнение една от кампаниите за опазване на околната среда (Earth Day) доведе до спестяване на около 56 MWh.

Табл. 3.4 Обобщение на интерфейсите към уеб услугата *Appliance Power Control*

Interfaces	Supported operations	Implementation
AirconStateMgmt	createState	HTTP POST
	retrieveState	HTTP GET
	modifyState	HTTP PUT
	deleteState	HTTP DELETE
StateActionMgmt	createStateAction	HTTP POST
	executeStateAction	HTTP PUT
	deleteStateAction	HTTP DELETE
TemparatureMgmt	createTemp	HTTP POST
	retrieveTemp	HTTP GET
	modifyTemp	HTTP PUT
	deleteTemp	HTTP DELETE
TempSubscMgmt	getTempSubs	HTTP GET
	updateTempSubs	HTTP PUT
RefTempSub	createRefTemp	HTTP POST
	retrieveRefTemp	HTTP GET
	modifyRefTemp	HTTP PUT
	deleteRefTemp	HTTP DELETE
RefTempNotification	refTemperature	HTTP PUT
	tempError	HTTP PUT
EconTempSub	createEconTemp	HTTP POST
	retrieveEconTemp	HTTP GET
	modifyEconTemp	HTTP PUT
	deleteEconTemp	HTTP DELETE
EconTempNotification	econTemperature tempError	HTTP PUT HTTP PUT



Фиг. 3.22 Консумация на електроенергия: (P2) традиционно, (P1) базирано на IoT график

Предложените модели и методи може да се използват в съвременни енергийни системи, насочени към по-ефективно използване на енергийните ресурси и намаляване на консумацията на електроенергия.

Четвърта глава. Изследване на модели и методи за самоконфигуриране на услуги в Интернет на нещата

В четвърта глава са изследвани модели и методи са самоконфигуриране на услуги. Автономните функции са свързани с откриване и разрешаване на нежелани взаимодействия между услуги. Ефектът от нежелано взаимодействия между услуги може да се представи като противоречие в логиката на съвместно използване на услугите. Методите за откриване и разрешаване на нежелани взаимодействия между услуги са илюстрирани с услуги в областта на мениджмънта на свързаността на устройства в комуникации от машинен тип. Моделите се основават на онтология, представена в авторска публикация **[A13]** и са описани с дескриптивна логика.

В терминологичното множество са въведени понятия, представящи състоянията, в които може да премине едно устройство от гледна тока на свързаност и местоположението му спрямо дефинирана област, както и факти, свързани с носещите ресурси:

- *disconnected* - устройството не е свързано;
- $connected_b$ - устройството е свързано и използва носещ ресурс *b*;
- $marginal_b$ - приетият сигнал от използвания носещ ресурс от страна на устройството е под предварително дефинираната стойност;
- *changeBearer*- устройството трябва да промени използвания носещ ресурс;
- *inArea* - устройството е в определена област;
- $preferred_b$ - носещият ресурс *b* е предпочитаният носещ ресурс за областта;
- $qosAcceptable_b$ - качеството на обслужване, предоставено от носещ ресурс *b* е задоволително;
- $qosUnacceptable_b$ - качеството на обслужване, предоставено от носещ ресурс *b* не е задоволително;
- $available_b$ - носещ ресурс *b* е наличен;
- $unavailable_b$ - носещ ресурс *b* не е наличен.

Преходите, които са използвани за промяна между състоянията на едно устройство, биват дефинирани под формата на роли:

- *signalDrop* - полученият сигнал от използвания носещ ресурс спада под определената от сървъра стойност;
- *signalRise* - полученият сигнал от използвания носещ ресурс се повишава над определената от сървъра стойност;
- *enter* - устройството постъпва в дадена област;
- *exit* - устройството излиза от дадена област;
- *qosDecreas* - качеството на обслужване на използвания носещ ресурс става неприемливо;
- *timerExpiry* - изтича времето, което е предоставено за подобряване на силата на сигнала;
- *connect* - устройството се свързва към мрежата;
- *disconnect* - устройството прекъсва свързаността си към мрежата.

Дефинирани са твърдения, които представят преходите в състоянията.

$$disconnected \sqcap available_b \sqsubseteq \exists connect.connected_b \quad (4.1)$$

$$connected_b \sqsubseteq \exists signalDrop.marginal_b \quad (4.2)$$

$marginal_b \sqsubseteq \exists signalRise.connected_b$ (4.3)

$marginal_b \sqsubseteq \exists timerExpiry.changeBearer$ (4.4)

$changeBearer \sqcap available_b \sqsubseteq \exists connec_b.connected_b$ (4.5)

$changeBearer \sqcap unavailable \sqsubseteq \exists disconnect.disconnected$ (4.6)

$disconnetced.connected_b \sqsubseteq \exists disconnect.disconnected$ (4.7)

Дефинирани са изрази, които описват мобилността на устройството. Устройството може да влезе в предварително дефинирана област и да използва предпочитания носещ ресурс.

$\neg inArea \sqsubseteq \exists enter.inArea \sqcap preferred_b$ (4.8)

$inArea \sqcap preferred_b \sqsubseteq \exists exit.\neg inArea$ (4.9)

Нека с DEV бъде означено множеството от всички устройства. Посредством CMS са означени всички състояния S_i от модела за управление на свързаността. Аксиоматичното множество съдържа твърденията, които представят първоначалното състояние на всяко устройство.

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

За да бъде изразен фактът, че всяко устройство в точно определен момент е в точно едно състояние, се използва твърдението:

$\top \sqsubseteq \neg(\sqcup_{d1,d2 \in CMS,\ d1 \neq d2}(s1 \sqcap s2)) \sqcap (\sqcup_{d \in CMS} s)$

Състоянията на устройствата се променят посредством действия, дефинирани под формата на функции. Функцията $Func_{CMS}$ за дадено състояние съответства на възможните преходи в модела на управление на свързаността. Фактът, че всяко устройство може да промени състоянието на свързаност само чрез определено действие, е представен чрез следното твърдение: за $\forall\ s \in CMS$, и $\forall\ R \notin Func_{CMS}(s)$, $s \sqsubseteq \forall R.s$.

Услугите биват моделирани като преобразувания на базата знания, използвайки контекста C[φ] като подформула φ на всяка формула ψ.

Услугата за „избор на носещ ресурс, базиран на местоположението“ (Location-based Bearer Selection-LBS) приема, че съществува предварително дефинирана географска област, в която трябва да се използва предпочитан носещ ресурс. Логиката на LBS услугата моде да се дефинира чрез следните твърдения:

$C_1[LBS \sqcap disconnected \sqcap available_b \sqcap inArea \sqcap preferred_b] \sqsubseteq \exists connect.C_2[connected_b \sqcap inArea \sqcap preferred_b]$ (4.10)

$C_3[\neg LBS \sqcap disconnected \sqcap available_c \sqcap inArea \sqcap preferred_b] \sqsubseteq \exists connect.C_4[connected_c \sqcap inArea \sqcap preferred_b]$ (4.11)

$C_5[LBS \sqcap disconnected \sqcap \neg available_b \sqcap inArea \sqcap preferred_b] \sqsubseteq C_6[disconnected]$ (4.12)

$C_7[LBS \sqcap connected_b \sqcap available_c \sqcap inArea \sqcap preferred_c \sqsubseteq \exists connect.C_8[connected_c]$ (4.13)

$C_9[\neg LBS \sqcap connected_b \sqcap available_c \sqcap inArea \sqcap preferred_c \sqsubseteq C_{10}[connected_b]$ (4.14)

$C_{11}[LBS \sqcap connected_b \sqcap \neg available_c \sqcap inArea \sqcap preferred_c] \sqsubseteq \exists disconnect.C_{12}[disconnected]$ (4.15)

$LBS \sqsubseteq \neg(connected_c \sqcap inArea_a \sqcap \neg available_b \sqcap preferred_b)$ (4.16)

Услугата за „избор на носещ ресурс, базиран на качеството на обслужване“ (Quality-based Bearer Selection-QBS), от своя страна изисква промяна, в случай че качеството на обслужване към използвания носещ ресурс спадне под предварително дефинираните стойности. Логиката на QBS услугата се дефинира чрез следните твърдения:

$C_1[\neg QBS \sqcap connected_b] \sqsubseteq \exists qosDecrease_b.C_2[connected_b]$ (4.17)

$C_3[QBS \sqcap connected_b] \sqsubseteq \exists qosDecrease_b.C_4[connected_b \sqcap qosUnacceptable_b]$ (4.18)

$C_5[QBS \sqcap connected_b \sqcap qosUnacceptable_b \sqcap available_c] \sqsubseteq \exists connect.C_6[connected_c]$ (4.19)

$C_7[QBS \sqcap connected_b \sqcap qosUnacceptable_b \sqcap unavailable]$

$\sqsubseteq \exists disconnect.C_8[disconnected]$ (4.20)

$QBS \sqsubseteq \neg(connected_c \sqcap qosUncceptable_c)$ (4.21)

Методът за автоматично откриване на взаимодействия между услуги се основава на разсъждения върху логическите твърдения. Методът е илюстриран с откриване на взаимодействия между базиран на местоположението и на качеството на обслужване избор носещ ресурс, и откриване взаимодействия между базиран на текущ баланс и скорост на предаване на данни избор на носещ ресурс, описани съответно в авторски публикации **[A14]** и **[A15]**.

Използва се табличен метод за стандартно извеждане на заключение от базата знания.

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 $C \sqsubseteq D \in T$

Фиг. 4.1 Табличен метод на репрезентация

Откриването на взаимодействия между LBS и QBS услугите функционира като се приложи AND към първоначалната формула, което резултира в четири отделни случая, а именно:

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

$disconnected \sqcap \neg available_b \sqcap available_c \sqcap inArea \sqcap preferred_b$

$\sqcup disconnected \sqcap \neg inArea \sqcap available_c \sqcup disconnected \sqcap unavailable) \rangle$

В случай $disconnected \sqcap available_b \sqcap inArea_a \sqcap preferred_b$ и прилагайки:

1.1 KB към правило (4.10) се получава:

〈ε | s_0: ¬*disconnected*⊔¬*available*$_b$⊔¬ *inArea*⊔¬ *preferred*$_b$ ⊔
∃*available*$_b$.(*connected*$_b$⊓*qosAcceptable*$_b$)〉

1.2 OR се получава разклонението:

1.2.1 〈0 | s_0: ¬*disconnected*}, което е затворено поради появата на {0 | s_0 : *disconnected*} в дадения сегмент по-рано.

1.2.2 〈0 | s_0: ¬*available*$_b$〉 (затворено).

1.2.3 〈0 | s_0: ¬*inArea*〉 (затворено).

1.2.4 〈0 | s_0: ¬*preferred*$_b$〉 (затворено).

1.2.5 〈1 | s_0: ∃*connect*$_b$.(*connected*$_b$⊓*inArea*) 〉

1.3 SOME се получава: 〈1 | s_0 *connect* s_1: (*connected*$_b$⊓*inArea*⊓*preferred*$_b$) 〉

1.4 KB към правило (4.18) се получава:

1.4.1 〈1 | s_0 *connect* s_1: ¬(*connected*$_b$⊓*inArea*⊓*preferred*$_b$)⊔
∃*qosDecrease*$_b$.(*connected*$_b$⊓*qosUnacceptable*$_b$⊓*preferred*$_b$)〉 , а след OR:

1.4.2 10 | s_0 *connect* s_1: ¬(*connected*$_b$⊓*inArea*⊓*preferred*$_b$)〉 (затворено).

1.4.3 〈11| s_0 *connect* s_1: ∃*qosDecrease*$_b$.(*connected*$_b$⊓*qos*
Unacceptable$_b$⊓*inArea*⊓*preferred*$_b$)〉 и прилагайки SOME
〈11 | s_0 *connect* s_1 *qosDecrease*$_b$ s_2:
connected$_b$⊓*qosUnacceptable*$_b$⊓*inArea*⊓*preferred*$_b$)〉 :

1.5 KB към правило (4.19) резултира в:

1.5.1 〈110 | s_0 *connect* s_1 *qosDecrease*$_b$ s_2: ¬*connected*$_b$⊓
qosUnacceptable$_b$⊓*inArea*⊓*preferred*$_b$ 〉 (затворено).

1.5.2 〈111 | s_0 *connect* s_1 *qosDecrease*$_b$ s_2: ∃*connect*.*connected*$_c$⊓*inArea*⊓*preferred*$_b$〉 , а след прилагането на SOME се получава 111 | s_0 *connect* s_1 *qosDecrease*$_b$ s_2 *connect* s_3: *connected*$_c$⊓*inArea*⊓*preferred*$_b$〉 , което противоречи на LBS⊑¬(*connected*$_c$⊓*inArea*⊓*preferred*$_b$).

2. В случай *disconnected*⊓¬*available*$_b$⊓*available*$_c$⊓*inArea*⊓*preferred*$_b$ устройството остава с прекъсната свързаност, както е в правило (4.12).

3. В случай *disconnected*⊓¬*inArea*⊓*available*$_c$ прилагайки:

3.1 KB към правило (4.1) и елиминирането на затворените случаи резултира в:

3.1.1 1 | s_0: ∃*connect*$_c$.(*connected*$_c$⊓¬*inArea*) 〉 към което след прилагане на SOME се получава: 1 | s_0 *connect* s_1: *connected*$_c$ ¬*inArea*) .

3.2 Последователно KB, OR и SOME към правило (4.8) се получава:

3.2.1 1 | s_0 *connect* s_1 *enter* *connected*$_c$⊓*inArea*⊓ *preferred*$_b$⊓*available*$_b$) 〉 , за което се наследява правило (4.13) и отново се прилага KB, OR и SOME с резултат 〈1 | s_0 *connect* s_1 *enter* s_2 *connect* s_3: *connected*$_b$⊓*inArea*⊓ *preferred*$_b$⊓*available*$_b$) 〉

3.2.2 〈1 | s_0 *connect* s_1 *enter* *connected*$_c$⊓*inArea*⊓*preferred*$_b$ ⊓¬*available*$_b$) 〉 , за което се извлича *connected*$_b$⊓¬*available*$_c$⊓*inArea*⊓*preferred*$_c$⊑∃*disconnect*.*disconnected* и прилагайки KB, OR и SOME се получава 〈1 | s_0 *connect* s_1 *enter* s_2 *disconnect* s_3: *disconnected*〉

3.3 Към $\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ connect\ s_3{:}connected_b \sqcap inArea \sqcap preferred_b \sqcap available_b) \rangle$ биват приложени същите стъпки както при 1.1.5, резултирайки в $\langle 1 \mid s_0\ connect\ s_1\ enter\ s_2\ connect\ s_3\ qosDecrease_b\ s_4\ connect\ s_5{:}\ connected_c \sqcap inArea \sqcap preferred_b \rangle$, което от своя страна противоречи на LBS.

4. В случай $disconnected \sqcap unavailable$ устройството остава с прекъсната свързаност.

Резултатът е затворено дърво, което означава, че $\delta_{QBS}(\delta_{LBS}(CMS))$ взаимодейства при активирането на {QBS}∪{LBS}.

Важно е да се отбележи, че взаимодействието между услугите може да бъде засечено автоматично чрез програмиране на алгоритъма. Предложен е метод за разрешаване на взаимодействието с услугите с използване на политики, който позволява самоконфигуриране на услугите. Автономните функции са свързани с автоматично откриване и разрешаване на нежелани взаимодействия между услуги и автоматичното договаряне на услугите, предоставяни от различни доставчици на услуги, за да се осигури адаптивна непрекъсваемост на услугите за крайните потребители

Заключение и основни приноси

В дисертационен труд са изследвани модели и методи автономен мениджмънт на устройства в комуникации от машинен тип. Анализът на проблемите в областта и прегледът на научните публикации потвърждават изводите, направени в първа глава и обосновава дефинираните задачи на научните изследвания. Предложените модели и методи са опит да се вгради автономно поведение в системите за мениджмънт на устройства с Интернет свързаност, като по този начин се улесни поддръжката на жизненият им цикъл.

Основните приноси на дисертационния труд могат да се обобщят по следния начин:

1. Изследвани са методи за мениджмънт на устройства. Проектирани са модели за регистриране на устройства, мениджмънт на свързаността, мениджмънт на неизправности и управление на качеството на обслужване. Моделите са описани формално и са верифицирани. Резултатите от изследванията може да се използват в системи за мениджмънт на устройства, като по този начин да улеснят процесите при поддържане на жизнения цикъл на системата. Предимствата на използването на предложените модели в сравнение със съществуващите, специфични такива са, че са базирани на широко приета рамка за мениджмънт, която е технологично независима и процедурите по мениджмънт могат да бъдат автоматизирани. Специфичните научни, научноприложни и приложни приноси включват:
   - Синтезиране на модели, отразяващи регистрационния статус на устройство, разширени с конфигурационен профил за мониторинг и докладване, и опресняване на софтуерната версия;
   - Разработване на метод за дефиниране на уеб услуги посредством REST архитектури за управление на неизправности, като са идентифицирани общи операции за уеб услуги, които са проектирани върху HTTP методи;

- Синтезиране на модели за мениджмънт на свързаността на устройства, които включват наблюдение на параметрите на свързаността, избор на носещ ресурс и конфигуриране на точка за достъп;
- Разработен е алгоритъм за избор на най-добрия носещ ресурс. Моделирано е поведението на автономен агент за мениджмънт на свързаността, който взима решение на база на дефинирани политики.
- Идентифицирани са изискванията към управление на качеството на обслужване в комуникации от машинен тип и е създаден модел на автономен агент за управление на качеството на обслужване.

2. Предложен е структурен метод за синтезиране на семантични данни в областта на домашни системи за енергийна ефективност. Метод включва моделиране на данни и дефинирането на интерфейси за уеб услуги. Семантичната анотация на данните дава знание за тяхната организация и съдържание и се използва при изграждане на автономни системи. Предложените модели и методи може да се използват в съвременни енергийни системи, насочени към по-ефективно използване на енергийните ресурси и намаляване на консумацията на електроенергия. Специфичните научни, научноприложни и приложни приноси включват:
   - Синтезиран е модел на семантична информация при управление на електро-енергията в едно интелигентно домакинство;
   - Предложен е метод за структуриране на ресурси, свързани с интелигентното отчитане на електроенергия. Методът позволява извличане на информация за потреблението на електроенергия и управление на домашни електрически уреди.
   - Разработен е метод за дефиниране на уеб услуги, които могат да се използват за управление на мощността на електрически уреди. Методът включва дефиниране на типове данни и интерфейси;
   - Разработен е метод за дефиниране на RESTful услуги, които могат да се използват за наблюдение и управление на потреблението на електроенергия. Методът включва дефиниране на типове данни и интерфейси.
3. Изследвани са модели и методи са самоконфигуриране на услуги. Автономните функции са свързани с автоматично откриване и разрешаване на нежелани взаимодействия между услуги. Разработен е метод за откриване на нежелани взаимодействия между услуги. Методът се базира на формално описание на логиката на услуги чрез твърдения в база знания и търсене на противоречия при въвеждане на описание на нова услуга. Предложеният метод позволява самоконфигуриране на услуги с използване на политики чрез автоматичното договориране на услугите, предоставяни от различни доставчици на услуги, за да се осигури адаптивна непрекъснатост на услугите за крайните потребители. Специфичните научни, научноприложни и приложни приноси включват:
   - Предложен е метод за формално описание на услуги за управление на свързаността в комуникациите от машинен тип посредством дескриптивна логика.
   - Разработени са модели на услуги за избор на носещ ресурс, базирани на местоположението, качеството на обслужване, текущия баланс и скоростта

на предаване на данни. Моделите са описани формално посредством дескриптивната логика.

- Синтезиран е метод за откриване на взаимодействия между услуги. Веднъж след като е засечено нежелано взаимодействие, то може да бъде избегнато посредством прилагането на политики и правила, свързани с въвеждане на приоритет на услуги.

Авторски публикации, свързани с изследванията в дисертационния труд

[A1] **Nikolov, A.** Challenges Faced by the Autonomous Management in Future Internet, Proc. of National Conference TELECOM 2016, Sofia, Bulgaria, pp.1-8. ISSN: 1314-2690

[A2] **Nikolov, A. "**Autonomous management issues in Future Internet", Journal Electrotechnica & Electronica, E+E, vol.51, No. 7-8/2016, pp.24-30. ISSN: **0861-4717**

[A3] Pencheva, E., **A. Nikolov,** I. Atanasov, K. Nikolova, "Modeling M2M Device Registration Status", Proc. of IFIP International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, PEMWN, Paris, France, 2016, **Indexed in Scopus.** ISBN: 978-1-5090-2671-5

[A4] Atanasov, I., **A. Nikolov**, E. Pencheva, „Design Aspects of Web Services for M2M Device Fault and Performance Management", Proc. of Int. Conf. TELSIKS, 2015, pp.289-292, **Indexed in SCOPUS**. ISBN: 978-1-4673-7514-6

[A5] **Nikolov, A.**, I. Atanasov, E. Pencheva. "Formal Verification of Connectivity Management Models in M2M Communications," Proc. of IEEE Int. Black Sea Conference on Communications and Networking, BlackSeaCom'2016, Varna, Bulgaria, 2016, pp.1-4, **Indexed in SCOPUS.** E-ISBN: 978-1-5090-1925-0

[A6] Atanasov, I., **A. Nikolov**, E. Pencheva. "Autonomous Agent Model for Connectivity Management in M2M Communications", Proc. of International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST, 2016, Ohrid, Macedonia, pp.1-4. ISBN: 978-9-9897-8678-5

[A7] Atanasov, I., E. Pencheva, **A. Nikolov**. "Study on Generic Functionality for Quality of Service Control on M2M communications," Proc. of IEEE International Conference BlackSeaCom, 2016, Varna Bulgaria, pp. 1-5, **Indexed in Scopus.** E-ISBN: 978-1-5090-1925-0

[A8] Atanasov, I., **A. Nikolov,** E. Pencheva, „Ontology Based Data Model for Power Control in Smart Homes", Proc. of International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST, 2015, Sofia, Bulgaria, pp.6-9. ISBN: 978-6-1916-7182-3

[A9] Atanasov, I., **A. Nikolov**, E. Pencheva. "An Approach to Transform Internet of Things Data into Knowledge," Int. Journal of Embedded Systems, Special issue on: "Advanced Mobile Cloud Computing and Internet of Things Systems and Networking," ISSN 1741-1068, 2017, vol.9, no.5, pp.401-412 **(SCOPUS SJR 0.26)**.

[A10] Atanasov, I., **A. Nikolov**, E. Pencheva, R. Dimova, M. Ivanov. "An Approach to

Data Annotation for Internet of Things," Int. Journal of Information Technology and Web Engineering, 2015, vol.10, No.4, pp.7-25, ISSN 1554-1045 **(SCOPUS - SJR 0.147)**.

[A11] Atanasov, I., **A. Nikolov**, E. Pencheva., "Reducing Energy Consumption by Using Smart Metering Intelligent Systems", Cybernetics and Information Technologies, vol.16, No 2, pp. 113-124, DOI: 10.1515/cait-2016-0024, ISSN: 1314-4081, (**SCOPUS SJR: 0.203**).

[A12] Atanasov, I., **A. Nikolov,** E. Pencheva., „Design of REST Web Services for Energy Management in Smart Metering Information Systems", Annual Proceedings of Technical University of Sofia, vol.5 issue 2, 2015, pp.9-18. ISSN: 1311-0829

[A13] Pencheva, E., I. Atanasov, **A. Nikolov**, R. Dimova, "Ontology and Reasoning on Device Connectivity," Proc. of International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST, 2017, Nish, Serbia, pp.1-4.

[A14] **Nikolov, A.,** E. Pencheva, I. Atanasov, K. Nikolova, "A Case of Service Interaction in M2M Device Management," Proc. of IFIP and IEEE International Conference on New Technologies, Mobility & Security, 2016, Larnaca, Greece, pp.1-5, **Indexed in Web of Science and Scopus**. E-ISSN: 2157-4960

[A15] Pencheva, E., **A. Nikolov**, I. Atanasov, K. Nikolova, "Modeling Service Interaction in M2M Connectivity Management," Global Wireless Summit GWS'2016, Aarhus, Denmark, pp.1-5, **Indexed in Web of Science and Scopus**. ISBN: 978-87-93609-30-3

Изследванията, представени в дисертационния труд са в рамките научноизследователски договор ДН 07/10 от 15.12.2016 г. на тема „**Изследване на архитектури, модели и методи за автономен мениджмънт в интернет на бъдещето**", финансиран от Фонд Научни Изследвания към Министерство на образованието, Република България.

Anastas Nikolov

RESEARCH ON METHODS AND MODELS FOR AUTONOMOUS MANAGEMENT IN INTERNET OF THINGS

Abstract

The Internet of Things (IoT) is an evolutionary step in the development of Internet technologies. IoT covers communications between intelligent objects that connect the digital world to the physical one. IoT communications set new network architecture requirements based on autonomous communication between sensors/activators and the corresponding application on the Internet. The doctoral thesis explores models and methods for autonomous management of devices with Internet connectivity in machine type communications. The first chapter focuses on analysis of some features of autonomous management in Future Internet. The review of the research in the field makes it possible to identify the problems and to formulate the main aim and tasks of the thesis i.e. to study the capabilities for autonomous management in machine-type communications and to suggest new models and methods for self-configuring and self-optimization of devices and IoT applications. The research in the second chapter is regarding to the device management methods. Models of device registration, connectivity management, performance management and service quality control have been designed, formally described and verified. Research results can be used in device management systems, and thus facilitating processes while maintaining the system life cycle. The benefits of using the proposed models, compared to the existing ones, are that they are based on a widely accepted management framework that is technologically independent and the management procedures can be automated. The development of models and methods for autonomous management can be aided by the synthesis of semantic information. A structural method for synthesizing semantic data in the field of home energy efficiency systems is proposed in the third chapter. The method includes data modeling and definition of Web Service interfaces. The semantic data annotation gives knowledge about data organization and content, and is used in building autonomous systems. The proposed models and methods can be used in advanced energy systems aimed at more efficient use of energy resources and reduction of electricity consumption. Aspects of self-configuration of services are explored in the fourth chapter. Autonomous features are related to automatic detection and resolution of undesired interactions between services. A method for detecting service interactions has been developed. The method is based on a formal description of service logic and it is illustrated for services related to device connectivity management. The automatic resolving of service interaction using policies allows service self-configuration and provisioning of adaptive service continuity to the end users.