The Personalization of a Cloud Ecosystem: Adding Dimensions to Situational Awareness

Invited Paper

Gordana Velikic, Milan Bjelica, Nemanja Ignjatov, Mica Cetkovic
RT-TK Institute for Computer Based Systems
Serbia, Novi Sad

Ivan Kastelan
FTN, University of Novi Sad
Novi Sad, Serbia

***Abstract*—The lack of easy to use environment to connect and manage the things is one of the obstacles that hold back multimodal personalized use of Internet of Things. We present a novel idea and a cloud framework with original elements that enable development of ecosystem for the user tailored deployment of things in the context of a personalized situational awareness. The benefits of the infrastructure are fortified with several use case scenarios that provide an insight to the platform's innovative possibilities.**

Keywords-Cloud, Internet of Things, Situational Awareness

I. Introduction

Internet of Things (IoT) [1] paradigm has been extensively promoted in recent years, however, the overall impression is that the number of end-user applications benefiting on the generic IoT platforms, such as Xively [2] or lately Apple's HomeKit [3] is still low.

Several problems are holding back IoT to evolve a step further from the basic eco-system of things. The major problem is lack of easy to use environment to connect and manage the things, and to navigate through the sea of data to extract desired user-centric information. Instead of searching through data and configuring devices, users are rather interested in gaining an easy access to information and devices, and to easily tailor parameters of applications according to their needs.

This leads us to situational awareness concept that can be easily utilized with easy to use user tailored platform to create ad-hoc applications for use case scenarios in different domains of interest, such as but not limited to security, wellness, or ambient adjustment. Situational awareness refers to knowledge about the environment and its changes over time or some other predefined parameter, like for example energy consumption [4]. The situational awareness should not be confused with a context awareness which is mostly associated with location awareness and activity recognition [5]. Context awareness is mostly applied to cases where user's behavior needs to adapt to physical environment. Situation awareness models users' environments and help them to be aware of current situation. The concept embedded in our platform is that user's can define and personalize specific processed situational awareness captured by the system for novel uses, such as but not limited to: home energy consumption in security or housekeeping monitoring, guest behavior in wellness, or TV channel status in ambient settings or parental control. Novel elements – user-friendly script agents enable users to individually define situational awareness use case scenario and create a script agent that will execute desired application.

II. Personal Cloud Agent Framework

Each object in our surrounding can be a building block for an IoT platform. Organized hierarchically, some objects may be able to detect raw physical parameters, some can devise state of the environment based on rules, and the most complex objects can provide alerts and guidance [6]. In this paper a concept of a personal cloud agent framework is proposed, which allows the creation of scriptable cloud agents. Using aggregated information in a customer profile, processed collected raw data acquired from devices, or both with additional information from other accessible sources, the customer can use the scriptable cloud agents to create personalized applications in home settings, or to stimulate actions on various devices available around customers in diverse commercial applications or settings.

The scriptable agents are part of a self-growing cloud-based platform for different content, devices, and environments (context based applications that can evolve into awareness based applications). The user, either an individual or industry, can tailor scripts to adapt or create novel scenarios and uses. For example, a hospitality industry, e.g. hotels, can create a specific personal cloud agent to send suggestions related to customer wellness back to all devices and applications in physical realm of IoT. The intention behind the creation of such agent is to enhance customers' experiences, their well-being, and overall satisfaction during a visit to vacation venue.

The proposed framework provides a technical medium for creation of objects for user-specific ad-hoc uses, such as above mentioned personal agents which provide guidance in wellness. A generic agent platform for IoT allows the implementation of IoT applications in which agents aggregate information sourced from various devices and protocols, and pass on the information among themselves to create target applications [7]. Script-based agents, written in Lua, allow for the implementation of complex logic and device self-management

[8]. Personal cloud agent framework is based on Insight IoT platform [8][9], which enables communication with end devices via TR-069 and XMPP and exposes appropriate API to above layers (see Figure 1).

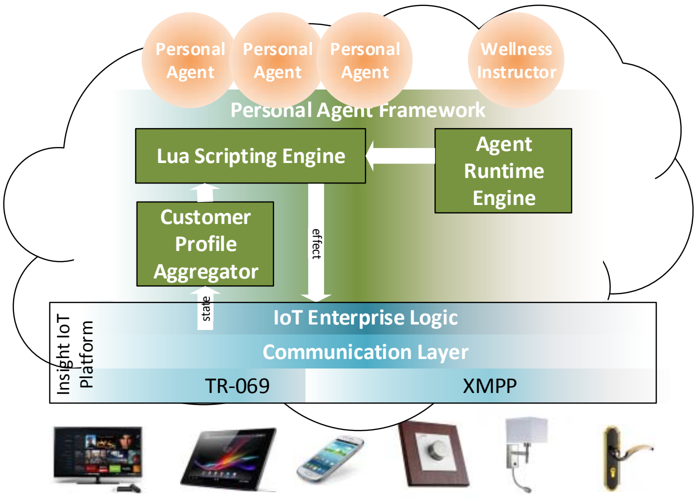

Figure 1. Personal Cloud Agent Framework Architecture: "personalized" for wellness aplication for use in hospitality industry.

Each device is presented with its *data model*, in the form of an object/parameter tree as defined in [10]. Each parameter can be set or read, and thus deploy a command to a device that either configures the device or queries its status. The *Customer Profile Aggregator* maps each device to an individual *customer profile*. Data model of each customer device is periodically processed to extract information relevant to the agent framework (as shown in Table I). *The Lua Scripting Engine* enables the execution of agents in the form of Lua scripts, while enabling extension APIs for Lua. This allows agents to access customer profiles and to manipulate with targeted device's data model from script language (e.g. to send deduced knowledge to a user application). The *Agent Runtime Engine* maintains the lifecycle of an agent instance, and links the script code of an agent with the customer profile and the device list that the agent is associated with. Additionally, the *Personal Agent Framework* provides an API for the top level web based applications, and allows them to manipulate the customer profiles and agents, and to make assignments.

TABLE I: INFORMATION RELEVANT TO A CUSTOMER PROFILE

Data model parameter	*Extracted information*
Outlet.1.Consumption.Value	Customer presence in the room
Dimmer.1.Mode	Customer activity / sleeping / bathroom
Time.CurrentLocalTime	Estimated daylight / Wellness availability / Expected biorhythm
STBService.Components. Service.CurrentService	Customer activity / Watched TV show / Customer mood
LAN.Stats.TotalBytesSent	Customer activity on phone / laptop

Personal agents are programmed in Lua scripting language, which is extended to use exposed Insight IoT APIs.

III. USE CASE SCENARIOS

A. *Wellness instructor*

Wellness instructors are tasked to detect user behavior which has bad impact on wellbeing (e.g. long watching of TV program or inappropriate content, bad habits such as long stay indoors), or to detect favorable conditions for well-being enhancement (e.g. nice weather, sauna or pool availability, diet menu suggestion, vitamin pill intake time). Based on these detections, these agents send appropriate messages to devices which may be able to communicate to customers (e.g. TV or mobile applications), or to apply corrective actions without human interventions, such as for example switching light off.

For the purpose of evaluation, a wellness instructor agent was programmed in Lua. Android set-top box was used as a Smart TV in a hotel room simulation, what would allow the customer in the room to review suggestions set by the agent in the specifically designed web-based wellness application, accessible via remote controller. The same set-top box was a member of IoT realm, disclosing information about the usage of TV. Additionally, the room was equipped with one smart outlet, two smart dimmer switches and one dimmable lamp, all reporting their state to the IoT platform. In addition to Android application, feedback from wellness instructor was also in the form of automatic control over lighting conditions in the room.

TABLE II: EXCERPT FROM THE DECISION TREE OF WELLNESS INSTRUCTOR

If this is observed…	*…then do the following*
Wellness working hours, using laptop for more than 2 hours, not watching TV, lights off	Suggestion: Swim in the pool to relax Action: Turn on lamp to 20% dim level to prevent eye fatigue
Just arrived in the room, evening, used light in the bathroom for 10 minutes, turns bathroom light off	Suggestion 1: Visit sauna before bed Suggestion 2: Go to bed earlier Action: Set bedtime reading lights
Morning, breakfast time, TV on time history recorded	Action: Turn TV on to music channel, set low volume, switch on wake up lighting scenario
Watching news on TV for more than 2 hours, late evening	Suggestion 1: Switch to comedy Suggestion 2: Go to sleep

An excerpt from the agent's decision tree is shown in Table II.

For the illustration purposes, the evaluation was performed regarding to functionality and to potential to enhance customer well-being. Functionality was validated in lab conditions with real devices and human user. The potential for well-being enhancement was evaluated through a simulated customer behavior with IoT devices based on the observation of family behavior in a hotel room during a vacation. According to the probability of customer hotel room occupancy during a day estimated as in [11], and the typical opening hours of hotel amenities (restaurant, pool, spa and gym), several time windows were identified that are appropriate for the use of wellness amenities (Figure 3). Most of the suggestions made by the wellness instructor agent (~90%) in the simulation matched these time windows. Also, wellness instructor was dimming lights to pleasant levels when a customer was believed to be sleeping, what matched to expected sleeping hours in 80% of test cases. Another possible scenario for this agent is to detect possible health hazard situation if the guest enters additional health data.

```lua

local step = 1000*5;
local limit = step*24;
local counter =0;

    setParameterValue("Device.EnergyManagementService.RoomSituation.1.TotalVacuumingTime",0,"key");
    setParameterValue("Device.EnergyManagementService.RoomSituation.2.TotalVacuumingTime",0,"key");
    setParameterValue("Device.EnergyManagementService.RoomSituation.3.TotalVacuumingTime",0,"key");
    setParameterValue("Device.EnergyManagementService.RoomSituation.1.VacuumedToday","false","key");
    setParameterValue("Device.EnergyManagementService.RoomSituation.2.VacuumedToday","false","key");
    setParameterValue("Device.EnergyManagementService.RoomSituation.3.VacuumedToday","false","key");

while counter <= limit do
   local room = math.random(1,1000);
   room = (room % 3) + 1;
   local vacuumpw1;
   local vacuumpw2;
   local vacuumpw3;

    vacuumpw1 = getParameterValue("Device.EnergyManagementService.EnergyMeter.1.CurrentConsumption");
    vacuumpw2 = getParameterValue("Device.EnergyManagementService.EnergyMeter.2.CurrentConsumption");
    vacuumpw3 = getParameterValue("Device.EnergyManagementService.EnergyMeter.3.CurrentConsumption");

    if(vacuumpw1 > "0") then
        local time1 = getParameterValue("Device.EnergyManagementService.RoomSituation.1.TotalVacuumingTime");
        setParameterValue("Device.EnergyManagementService.RoomSituation.1.TotalVacuumingTime",time1 + step,"key");
        setParameterValue("Device.EnergyManagementService.RoomSituation.1.VacuumedToday","true","key");
    end

    if(vacuumpw2 > "0") then
        local time2 = getParameterValue("Device.EnergyManagementService.RoomSituation.2.TotalVacuumingTime");
        setParameterValue("Device.EnergyManagementService.RoomSituation.2.TotalVacuumingTime",time2+step,"key");
        setParameterValue("Device.EnergyManagementService.RoomSituation.2.VacuumedToday","true","key");
    end

    if(vacuumpw3 > "0") then
        local time3 = getParameterValue("Device.EnergyManagementService.RoomSituation.3.TotalVacuumingTime");
        setParameterValue("Device.EnergyManagementService.RoomSituation.3.TotalVacuumingTime",time3+step,"key");
        setParameterValue("Device.EnergyManagementService.RoomSituation.3.VacuumedToday","true","key");
    end

    if((vacuumpw1 == "1000") and (vacuumpw2 == "1000")) then
           setParameterValue("Device.EnergyManagementService.CurrentAlerts","Room 1 and Room 2 is currently being vacuumed.","key");
    elseif((vacuumpw2 == "1000") and (vacuumpw3 == "1000")) then
       setParameterValue("Device.EnergyManagementService.CurrentAlerts","Room 2 and Room 3 is currently being vacuumed.","key");
    elseif((vacuumpw1 == "1000") and (vacuumpw3 == "1000")) then
       setParameterValue("Device.EnergyManagementService.CurrentAlerts","Room 1 and Room 3 is currently being vacuumed.","key");
    else
        setParameterValue("Device.EnergyManagementService.CurrentAlerts","","key");
    end


   sleep(step);
   counter = counter + step;
end
```

Figure 2. “Raw” situational awareness script for the use case scenario B.

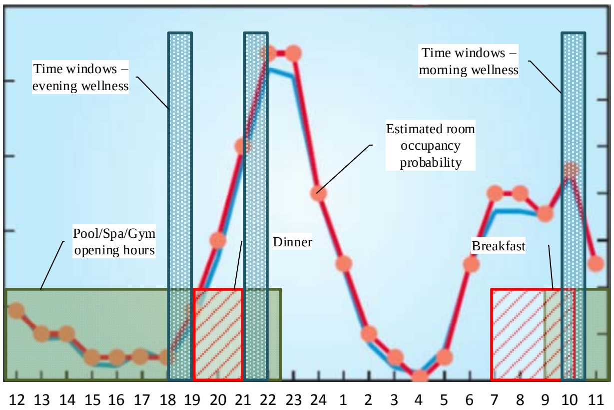

Figure 3. Time windows appropriate to use the hotel wellness amenities.

B. Alert of possible intruder, monitoring house keeping

Another example of situational awareness scenario demonstrates the utilization of a seemingly “useless” sensor, such as the smart outlet with energy measurement capability. In the scenario, it is possible to detect if a chore has been correctly accomplished for a given week. For example, a housemaid is tasked to perform vacuum cleaning of the whole home. Situational awareness script, as shown in Figure 2, enables the detection of the task fulfillment based on energy consumption from outlets in the vacuumed rooms. Assuming that only one vacuum cleaner is available and just one housemaid, script assumes that no two rooms can be simultaneously vacuumed and sets alerts accordingly. In the given script, energy consumption is being continuously acquired from smart outlets in each of the three rooms. If the consumption is non-zero, total vacuuming time for the respective room is being increased. If consumption is detected on more than one outlet at any given time, script alerts the homeowners of the inconsistency. In addition, homeowners are able to see final vacuuming times which are stored to output parameters, and therefore validate if the vacuuming was performed for each of the rooms in the adequate duration.

Quick Parameter View

Device Name	RoomMonitoring	Set
CurrentAlerts	Room 1 and Room 2 is current...	Set
CurrentConsumption	1000	Set
CurrentConsumption	0	Set
CurrentConsumption	1000	Set
VacuumedToday	true	Set
TotalVacuumingTime	70000	Set
TotalVacuumingTime	10000	Set
VacuumedToday	true	Set
TotalVacuumingTime	20000	Set
VacuumedToday	true	Set

Figure 4. Web application snap shot for the use case scenario B.

```
local step = 1000*5;
local limit = step*24;
local counter =0;

setParameterValue("Device.STBService.1.Components.UserActivityMonitoring.CurrentAlerts","","key");


while counter<=limit do
    setParameterValue("Device.STBService.1.Components.UserActivityMonitoring.CurrentTime",os.date("%c",),"key");
    local lastServiceChange = getParameterValueAndWait("Device.STBService.1.Components.UserActivityMonitoring.LastZapClk");

    print("Time elapsed since last service change "..os.time()-lastServiceChange);

    if((os.time()-lastServiceChange) > 4*60*60 ) then
        setParameterValue("Device.STBService.1.Components.UserActivityMonitoring.CurrentAlerts","Entering sleep mode in progress!");
    end

    sleep(step);
    counter = counter + step;

end
```

Figure 5. "Raw" situational awareness script for the use case scenario C.

Figure 4 shows the snapshot from a demonstration web application, which can be accessed by homeowners. In this snapshot, alerts can be accessed as well as current consumption on each of the outlets and, more importantly, a conclusion from the awareness script, stating if the room was finally vacuumed or not. Although a number of improvements with regard to edge cases and false diagnoses can be applied during the script refinement, the given example indicates the feasibility of situational awareness with the given platform even in the case of availability of only the most simplistic sensing. Examples of other uses with the same script may be intruder alert during vacation, or unexpected kid's party during parents night out. The alerts may be forwarded to phone or e-mail.

C. Ambient settings

The following situational awareness script enables the remote detection of idleness in TV watching. The script (see Figure 5) can efficiently detect if the channel was not changed for an excessive time (e.g. 4 hours), and suggest the sleep mode to the TV device via a specific parameter setup (as in Figures 6 and 7). The demonstration of the operation is shown in snapshots from the web application, showing the currently watched channel, last zapping time and alert field, indicating the sleep mode situation, if detected. Additional information from passive infrared sensor would further enhance this scenario, and enable more accurate detection of sleep mode, mitigating the edge case in which the user excessively watches one particular channel without zapping. Integration of situational awareness system with hybrid, connected devices, such as set-top boxes, is a natural step and the presented example shows one potential usage. Another possible scenario that can be built in this script is ambient mode settings for particular genre or channel.

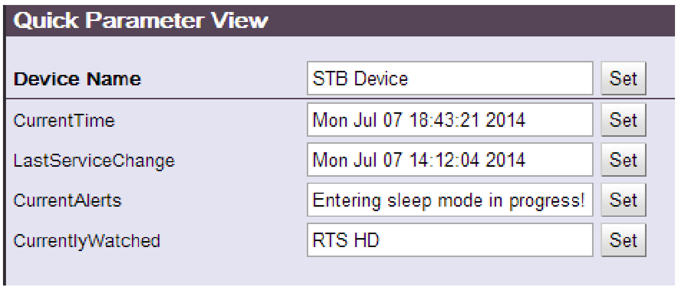

Figure 6. Web application snapshots for the use case scenario C. Given plot shows time span almost ten minutes and no alerts generated.

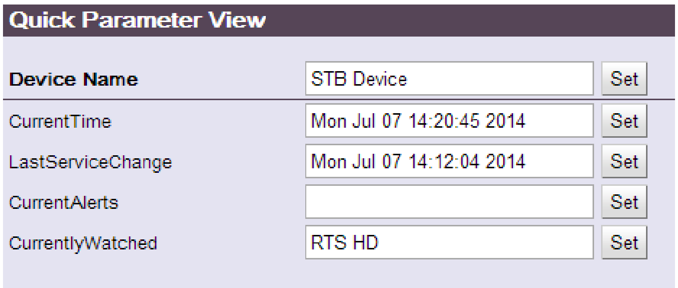

Figure 7. Web application snapshot for the use case scenario C. Given plot shows generated alert that will set preset actions since no activity is registered in defined time frame. In this case it is about four hours.

CONCLUSION

The presented cloud based platform cancels the need for user to have extra computational power, ads to wider accessibility, mobility, and lower maintenance cost for application developers. Modular and device agnostic approach makes the platform adaptable to future technology developments, extendible, and scalable. These are all features that most good platforms have. However, with our scriptable agent, it delivers an easy way for an individual customer to take control of personalization and turn data collected from day to day tasks, like cleaning, or home automation, into desired use cases without a need for professional technical help. Although the platform needs finishing touches, like intuitive and "fancy" graphical user interface to deploy commercially, it is ready and "fully loaded" to add new dimensions to situational awareness, as we showed in three examples.

REFERENCES

[1] K. Ashton, "That "Internet of Things" Thing," RFiD Journal 2009.

[2] Xively, https://xively.com/, 2014.

[3] Apple HomeKit Framework, https://developer.apple.com/homekit/, 2014.

[4] Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. In Proceedings of the human factors society 32nd annual meeting, Santa Monica, CA (Vol. 1, pp. 97–101).

[5] Schilit, B., & Theimer, M. (1994). Disseminating active map information to mobile hosts. IEEE Network, 8, 22–32.

[6] G. Kortuem, F. Kawsar, D. Fitton and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," *Internet Computing, IEEE,* vol. 14, no. 1, pp. 44-51, 2010.

[7] A. Inmaculada, M. Amor and L. Fuentes, "An agent platform for self-configuring agents in the Internet of Things," *Proceedings of ITMAS 2012*, pp. 65-78, 2012.

[8] M. Z. Bjelica, N. Ignjatov, I. Papp and N. Teslic, "Device Cloud Platform with Script-Based Agents for Anywhere Access Applications Development," *Proceedings of MIPRO,* 2014.

[9] M. Z. Bjelica, G. Golan, S. Radovanovic, I. Papp and G. Velikic, "Adaptive Device Cloud for Internet of Things Applications," *Proceedings of ICCE China*, 2014.

[10] "TR-106 Data Model Template for TR-069-Enabled Devices," *Broadband Forum*, Issue 1, Amendment 3, 2009.

[11] "Energy Conservation for Hotels, " *The Energy Conservation Center, Japan*, 2009.