2015 IEEE 12th International Conference on Autonomic Computing

Constructing Execution and Life-Cycle Models for Smart City Services with Self-Aware IoT

Masahide Nakamura
Graduate School of System Informatics, Kobe University
1-1 Rokkodai, Nada, Kobe 657-8501, Japan
Email: masa-n@cs.kobe-u.ac.jp

Lydie du Bousquet
Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
CNRS, LIG, F-38000 Grenoble, France
Email: lydie.du-bousquet@imag.fr

***Abstract*—Although various smart city projects are launched in all over the world, it is not obvious how to tailor the existing IoT and self-aware technologies for individual services, systematically. One of the reason is due to the lack of *common view* that can be used to investigate various smart city services across different domains. This paper proposes a *domain-neutral execution model* and an *integrated life-cycle model* of smart city services. We first identify essential activities for smart city services based on the *city-as-a-state-machine* concept. We then adopt goal-oriented thinking which clearly decomposes a goal and a means for each of the essential activities. By doing so, the proposed models can grasp essentials of any smart city service with domain-neutral activities and life cycles, while domain-specific parts can be varied by the means. Using the proposed models, we conduct a case study with smart car parking, where the proposed method compares the four different parking services. Finally, we develop ideas where and how the IoT and self-aware technologies can be applied effectively.**

I. Introduction

A *smart city* is a next-generation urban planning that aims to enhance performance of a city as well as wellbeing of the citizens using modern information communication technologies (ICT) [1][2]. Recently, many smart city projects are under way in all over the world to tackle social problems within individual cities [3]. In a smart city, various services and solutions are proposed for a wide range of *domains*, covering from the *hard* aspect of the city to the *soft* aspect of the city [1]. The hard domains exploit technologies to optimize energy, transport and buildings, while the soft domains develop people for better government, economy and society.

IoT and the self-aware autonomic systems are cutting-edge technologies, promising to achieve highly automatic, wide-range and scalable smart city services [4][5]. However, it is not yet quite obvious how to adapt these technologies to every instance of smart city service. The challenge lies in not only the great variety of the problem domains, but also the local contexts of the cities. Hence, most smart city projects have been conducted somehow in bottom-up and technology-driven fashions, without common definitions, shared bodies of knowledge, or even clear recognition by citizens [6].

In this paper, we explore a way to investigate any given smart city service systematically, from the viewpoint that where and how the IoT and self-aware technologies can be applied effectively. A promising approach is to have a *domain-neutral model* of smart city services. Although there are several existing studies (e.g., [3][7][8][9]) that characterize smart cities by a common set of metrics, the granularity is too coarse to investigate how individual services are operated.

In this paper, we propose two kinds of models that provide a common view of the smart city services. The first model is the *execution model*, which characterizes run-time behaviors of a given smart city service. Regarding a city as a *state machine*, we define every service as a sequence of actions that change the current state of the city to a “to-be” (i.e., ideal) state. Based on the city-as-a-state-machine concept, we identify four *key activities* necessary for the service to achieve successful state transitions: (1) state observation, (2) state interpretation, (3) action proposal, and (4) actions execution. Each activity is derived by goal-oriented thinking, which clearly decomposes its domain-independent goal and domain-specific means.

The second model is the *life-cycle model*, which characterizes a whole life-cycle of a smart city service. In addition to the run-time cycle covered by the execution model, the life-cycle model integrates two additional life-cycles: (a) deployment/adjustment cycle and (b) design/re-design cycle. By using the proposed models, we can grasp essentials of any smart city services with domain-independent activities and life-cycles. It gives service developers a *clear map* of how and where to apply their technologies. The proposed models also allow analysts to compare different services and solutions for the same goal with respect to the maturity and smartness.

To demonstrate the feasibility, we conduct a case study of smart car parking, where we compare different car parking services using the proposed model. Finally, we develop ideas where and how the IoT and self-aware technologies can be applied effectively to these smart parking services.

II. Preliminaries

A. Smart City and Services

A *smart city* aims to enhance the performance of the city as well as well-being of the citizens with a clever use of ICT. In general, the existing problems, the local contexts and the goal to be achieved vary from one city to another. Therefore, a wide variety of smart city projects are diffused in various *domains*. Neirotti et al. identified six major domains of smart cities [1]: (1) natural resources and energy, (2) transport and mobility, (3) building, (4) living, (5) government, (6) economy and people. They are ordered from *hard* domains to *soft* domains, according to the orientation to hard or soft resources. Each domain is decomposed further into sub-domains.

978-1-4673-6971-8/15 $31.00 © 2015 IEEE
DOI 10.1109/ICAC.2015.57

In this paper, a *smart city service* refers to any social activity in a smart city that directs the city to *ideal* status with a *specific goal*. For example, a green energy service may turn off public air-conditioning and lighting where nobody is around, in order to reduce the energy consumption. A smart parking service navigates a driver to the nearest free parking, in order to optimize the traffic for finding parking.

In literature, the terms "smart city" and "smart city service" are often used interchangeably. However, we clearly distinguish them, since a smart city (as a platform or infrastructure) can provide multiple smart city services (as applications) to cope with problems in different domains. Such distinction can be also seen in the latest projects such as ClouT [10].

B. IoT and Self-Aware Systems for Smart City Services

IoT (Internet of Thing) is a key enabling technology for smart cities, since a smart city service often interacts with the physical environment of the city. The networked sensors and actuators allow automatic data collection and operations, which help efficient execution of the service. Considering the management aspect of the smart city services, the concept of *self-aware systems* in autonomic computing [4] becomes more important. As a smart city service covers a wide area of the city or a huge number of entities, the complexity lies not only in the operation, but also in the installation and maintenance.

Thus, the IoT with the self-aware capabilities will make services with such large-scale configuration highly manageable by limited human effort. Gurgen et al. [5] presented the self-* properties especially important for CPS (cyber-physical systems): self-adaptation, self-organization, self-optimization, self-configuration, self-protection, self-healing, self-description, self-discovery, and self-energy-supplying. However, it is not yet obvious how to adapt and tailor these cutting-edge technologies to individual smart city services.

C. Goal and Challenge

In order to see how and where the existing technolgies can be applied, we aim to develop a method that can be used to investigate any given smart city services, systematically. Due to the wide variety of domains and the local contexts, smart cities and their services have been developed individually in technology-driven and bottom-up fashions, without common definitions or shared bodies of knowledge [1][2].

The challenge here is how to derive a *domain-neutral model*, which provides a common view of heterogeneous smart city services across different domains. For example, for both the green energy service and the smart parking service in Section II-A, the model should be able to provide the common means to undestand *how* and *for what* these services are developed, operated and maintained.

Indeed, there are several existing studies [3][7][8][9] that categorize smart cities by a common set of attributes (e.g., objectives, tools, stakeholders, business models, etc.) They are useful to obtain static profiles of individual cities. However, the granularity is too coarse to investigate dynamic aspects on how individual services are developed, operated and maintained.

III. Designing Domain-Neutral Model for Smart City Services

A. Approach Overview

The proposed method consists of three elements:

- the city-as-a-state-machine concept
- the service execution model
- the integrated life-cycle model

Firstly, the *city-as-a-state-machine concept* allows us to obtain a domain-neutral view of smart city services, In this concept, a smart city is defined as a state machine, which changes the state according to an action performed. A smart city service is defined as a sequence of actions which directs the state machine to an acceptable state. Secondly, the *service execution model* identifies four key activities, essential for a service to achieve the successful state transitions in the city state machine. Thirdly, the *integrated life-cycle model* defines the whole life-cycle of a service, including design/re-design and deployment/adjustment stages. The following subsections explain the details of the three elements.

B. Considering City as a State Machine

In Section II-A, we defined a smart city service as any social activity that directs the city to ideal status with a specific goal. Hence, it is natural to consider that a city has a *state*, which can be good or bad from the viewpoint of the service. Performing an *action* within the city can change the current state to another state, which may (or may not) direct the city closer to the ideal status. From this observation, we regard a given city as a *state machine*. A state of a city can be represented by a vector $[s_1 : v_1,\ s_2 : v_2,\ ...,\ s_m : v_m]$, where s_i is an (interesting) attribute and v_i is its value[1]. For instance, a state can be represented like `[CO2:440ppm, TotalPowerConsumption:3000MW, #ofCarsInHighWay:1234, AvgSpeedOfCars:15km/h]`. Turning off some public lighting may change the state by decreasing the value of `TotalPowerConsumption`.

Whether a given state is "ideal" or not depends on the goal of the service. The role of a smart city service is to change the current "as-is" state to the ideal "to-be" state with a clever use of ICT. Hence, for a given city C, we define a smart city service S as a sequence $[a_1, a_2, ...]$ of actions that changes the current state of C to an ideal state. Figure 1 shows a schematic representation of the city-as-a-state-machine concept. Note that this concept can be applied to any smart city domains.

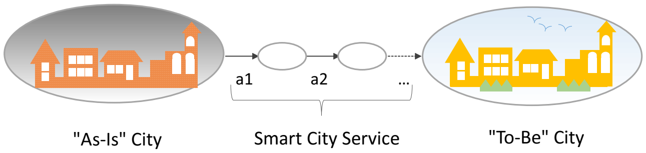

Fig. 1. City as a State Machine

[1] More theoretically speaking, the set of all possible states is defined by *parallel composition* of states of all entities involved in the city, where s_i and v_i respectively denote the state variable and value of i-th entity.

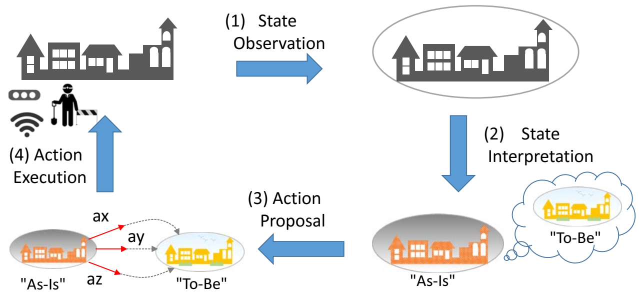

Fig. 2. Service execution model

C. Service Execution Model

We here identify four key activities necessary for a smart city service to achieve successful transitions from the “as-is” state to a “to-be” state. For each activity, we carefully consider its *goal* and *means* separately, so that the goal (used as a label of the activity) constructs the domain-neutral framework, while the domain-specific parts are varied by the means.

(1) State Observation: What a smart city service has to do first under the city as a state machine is to *observe the state* of the city. In other words, the service must obtain the vector of values of interesting attributes in some ways. Typical means to implement this activity include massive data collection with IoT, reference of open data, participatory sensing, crowd sourcing, (manual or automatic) meter readings, questionnaires, surveys and national census. No matter which method is taken, they are all for observing the current state of the city. We assume that this activity does not include any data processing for the obtained data.

(2) State Interpretation: The next activity for the service is to perform operations that allow the service or stakeholders to *interpret the state*, i.e., to evaluate how far (or close) the current state is towards the goal of the service. It is performed based on the values of the vector obtained in (1). The interpretation can use raw values of the state data, or applies aggregation, abstraction or projection to the raw data. One can exploit the emerging big data analysis for deriving a high-level context and knowledge, or use the latest stream mining to reason the real-time contexts. It can also be done based on human expertise and experience of domain experts. If the current state is shown to be acceptable with respect to the goal of the service, the following activities are skipped and the service goes back to (1).

(3) Action Proposal: When the current state is different from the ideal state, the service or stakeholders may *propose actions* that can get the city closer to the “to-be” state. Note that the actions to be proposed are not limited in automatic operations with the ICT. It is rather rare to fully automate a smart city service, considering the scale of the problem. Example actions include planning device operations, displaying appropriate suggestions to citizens, selecting technicians to send, encouraging citizens to do something, etc. If the “as-is” state requires a drastic change towards the “to-be” state, it is also important to propose new policies or regulations as actions. The action proposal requires high-level knowledge to understand the current state and derive the possible solutions. Technologies such as automated planning, machine learning, context-aware and personalization techniques can help partially to implement this activity.

(4) Action Execution: Finally, the service or stakeholders *execute the actions* within the city. The execution of the actions can give some effects to the city, changing the current state to another state. The subjects and the methods for the execution vary from one action to another. The remote operations of equipment may be executed automatically by sending commands to networked actuators. The fresh information may be displayed on variable-message signs in the roads, or in smart phones of citizens. The city maintenance work may be conducted by technicians. Of course, citizens themselves are major actuators who execute the actions. In this case, the ICT is used to *assist* citizens to execute the actions. Following this activity, the service goes back to (1) for continuous and sustainable improvement of the city state.

Figure 2 shows the proposed service execution model containing the four activities (1)–(4). The validity of the model is partially supported by the existing *MAPE-K model* [11], organizing a similar control loop with four phases: Monitor, Analyze, Plan and Execute. It was originally proposed for autonomic computing, where the control loop was managed automatically by an *autonomic manager* [4]. Our execution model can be seen as an adaptation of the MAPE-K model for the smart city domain. However, our model allows human intervention in the loop by emphasizing “for what” each activity exists, instead of considering “how” the loop should be implemented.

D. Integrated Life-Cycle Model

In addition to the run-time activities covered by the execution model, it is also important to consider how a smart city service is designed, deployed and maintained. This motivated us to consider a whole *life-cycle* of the smart city service. As mentioned in [6], a smart city should be sustained in the future, and must not end up with the short-term governmental policies. Therefore, the life-cycle must contain *iteration* to maintain and improve the service, where the iteration can be performed in an agile way across stages of design, deployment and run-time.

Hudert et al. [12] proposed a service-centric life-cycle model for electromobility systems. This model nicely covered our requirement. However, the run-time cycle was designed specifically for the electoromobility system. We have therefore adjusted the model for our purpose of the smart city.

Figure 3 shows the proposed *life-cycle model*, which integrates three levels of life-cycles: (a) *run-time cycle*, (b) *deployment/adjustment cycle*, and *(c) design/re-design cycle*. Most portion of the life-cycle model is the same as the original one in [12]. However, the run-time cycle is replaced with our service execution model presented in Section III-C.

The *run-time cycle* contains the service execution model with the four activities. These activities forms a cycle in order to get the city close to a “to-be” state, which is as explained in the previous sub-section. The run-time cycle starts when the service is successfully deployed. When the service does not work correctly, the run-time cycle moves to the adjustment process. If the service is no more maintained, the whole life-cycle terminates, as shown in the sink state.

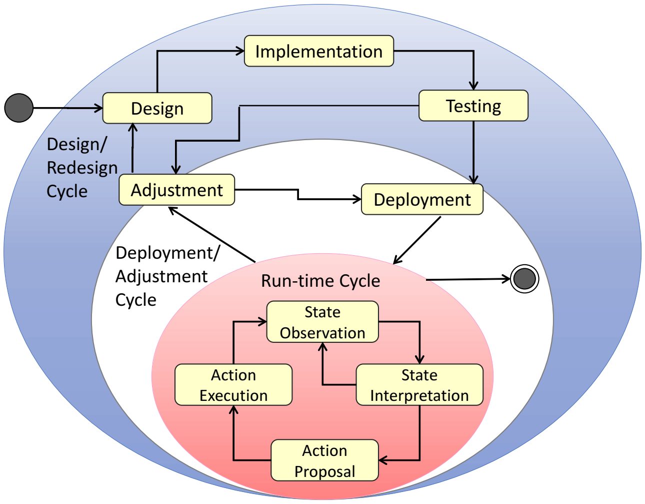

Fig. 3. Integrated life-cycle model

The *deployment/adjustment* cycle contains two processes: *deployment* and *adjustment*. The deployment process makes the smart city service ready to use in the city. Major activities include installation of hardware (sensors, equipment, device, etc.) to the city infrastructure, distribution and installation of software to citizens' smartphones, configuration of system parameters, service registration of things and citizens. Once the deployment is completed, the run-time cycle is initiated.

The adjustment process conducts *tuning* of the service without rebuilding the whole system. The process is executed when the service fails or requires maintenance. The adjustments include changing configuration parameters, expanding the system scale, repairing hardware, revising operational policies, periodic diagnoses. When the adjustment succeeds, the process moves to the deployment process to resume the service. If the adjustment fails for some reasons, the process moves to the design process to re-design the service.

The *design/redesign cycle* develops the service based on typical software engineering processes: *design*, *implementation* and *testing*. According to the agile practice, this cycle may be iterated through the adjustment process after testing. When all the test-cases are passed, it moves to the deployment process to install and deploy the service in the real environment. The design/redesign cycle is triggered again when the operated service cannot be recovered by the adjustment.

IV. Investigating Smart Parking with Models

To demonstrate the feasibility, this section investigates the existing *smart parking services* using the proposed models. We compare different parking services using the proposed models. We then evaluate where and how the IoT and self-aware technologies can be applied effectively to these services.

A. Smart Parking Services

The smart parking is one of most typical smart city services operated in many cities. The goal of the service may vary slightly from one city to another. However, it mainly aims at helping drivers to find a vacant parking space easily. This decreases energy consumption and pollution during circling for parking. It can also contribute to controlling traffic density and collecting taxes. In the following, we introduce four smart parking services actually operated in different cities.

1) Grenoble, France: Several public parking lots in Grenoble exploit smart parking systems, operated by Vinci Park [13]. A toll machine installed in a parking lot counts the number of cars incoming to and outgoing from the parking lot, which calculates the total number of free slots. The number of free slots in the parking lot is advertised in a variable message sign in the road. The occupancy status of parking lots in different locations are displayed on a map on the web, so that drivers can search location and availability with their smart phones.

2) Santander, Spain: Santander smart city testbed [14] exploits 375 magnetic sensors (connected with Libelium Waspmort [15]) for outdoor parking area management. Each of the sensors is buried under the asphalt of each parking slot, in order to detect the presence of a car on the slot. The sensors provide fine-grained occupancy information, where the driver can see the availability of every single slot. Based on the information, ten variable message signs collaborate to guide drivers to free slots. The system also provide the information through the web or the smart phone app.

3) San Francisco, USA: SFPark [16], smart parking in San Francisco, also employed wireless sensors to detect parking space occupancy in real time. The information is available through road message signs, the Web, and the smart phone app. What is unique here is that SFpark uses *demand-responsive pricing* in order to balance the occupancy. The parking rate becomes more expensive (or cheaper), depending on when and where the space is more occupied (or vacant, respectively). Based on the collected data, SFpark periodically adjusts meter and garage pricing up and down to match demand.

4) Brussels, Belgium: Carambla [17] operates a quite unique parking service in Brussels and other cities in Belgium. Instead of managing parking space with smart systems, Carambla provides a *marketplace* of the parking. An owner of a parking space (public or private) registers detailed information about the parking space to rent (owner's profile, location, price, size, etc). On the other hand, a driver also registers information of profile, owned car, payment, etc. Using the designated app, the driver chooses a parking space, and the app navigates the driver to the space. Once arriving at the space, the driver manually tells the system the start of parking using the app. When leaving the space, the driver tells the system the termination. The fee is paid to the owner via Carambla, according to the time of parking.

B. Comparing Different Services Using Proposed Method

We here investigate the above four different services using the proposed models. It would be interesting to see how they are different in which aspect based on the common criteria provided by the models. Specifically, for each smart parking service, we describe means of how to achieve the goal of each process (or activity). For every task in the means, we carefully describe the *subject* to clarify who is responsible to the task. The subject may be a human or a system, including the driver, the system, the app, the equipment, or so on.

TABLE I. COMPARISON OF FOUR DIFFERENT SMART PARKING SERVICES

Life-Cycle	Process	Grenoble, France	Santander, Spain	San Francisco, USA	Brussels, Belgium
Supplementary Service Goal		N/A	Service provides fine-grained and real-time occupancy information.	Service provides fine-grained and real-time occupancy information. Service creates the right level of parking availability.	Service creates business chances to space owners. Service creates more spaces to drivers without investment.
Run-time Cycle	State Observation	Toll machine counts # of incoming and outgoing cars.	Wireless sensor detects presence of car on every slot.	Wireless sensor detects presence of car on every slot.	Driver manually inputs start and end of parking with smart phone app.
	State Interpretation	System calculates total # of free slots in each parking lot. System gathers the number for all parking lots.	System generates occupancy information for the slot. System gathers the information for all slots in area.	System generates occupancy information for the slot. System gathers the information for all slots in area. System updates price of slot every hour of day.	System checks every parking offer and usage. System generates occupancy information for the used parking.
	Action Proposal	System advertises # of free lots on road sign and web portal.	System provides map of occupancy on web and smart phone app. Road signs display a way that guides all incoming drivers to free slots.	System provides real-time occupancy map on smart phone app. System suggests price information.	System suggests several parking deals based on the request.
	Action Execution	Driver sees the sign, decides a parking lot, and parks the car.	Driver follows the signs, finds a parking slot, and parks the car.	Driver sees info and price on app, decides and finds a parking slot, and parks the car.	Driver sees the suggestion, makes parking contract with app. App navigates driver to the parking. Driver parks the car.
Deployment/ Adjustment Cycle	Deployment	City installs a toll machine in the gate of a parking lot. City installs a variable message sign to road. City configures them to connect to system.	City installs wireless sensors to every parking slot. City installs variable message signs to roads. City configures them to connect to system. Driver may install smartphone app.	City installs wireless sensors to every parking slot. City installs smart parking meter to each block. City configures them to connect to system. Driver installs smartphone app.	Owner registers profile and space info. Driver makes profile and car info. Driver installs smartphone app.
	Adjustment	City maintains parking lot, toll machine, road sign, and Web portal.	City maintains parking space, wireless sensors, road signs, Web and app.	City maintains parking space, wireless sensors, smart meters, Web and app. City adjusts parking rates according to the demand and response.	Carambla maintains Web and app. Parking owner maintains the space. Carambla maintains customer relation.

Table I shows the result of the investigation, comparing the four different parking services. In the table, each column shows description of a service. The second row describes *supplementary goal* of service, which is a secondary goal or value specifically focused by the service. The four services are compared in each process of the run-time cycle and deployment/adjustment cycle. Since we were not able to know how these services were developed, we did not fill the description of the design/re-design cycle.

We first look at the run-time cycle. As seen in the action execution, the primary goal of every service is to allow the driver to park the car. However, each service aims at the further supplemental goal, which is different from each other. The fine-grained and real-time occupancy information is the value of Santander and San Francisco services. This value is achieved by the wireless parking sensors deployed in individual parking slots. The action proposal of Santander includes the guiding feature with the road signs, which prevents the drivers from taking wrong ways. San Francisco achieves the right level of parking availability by proposing the demand-responsive dynamic pricing. The availability creates the crucial value especially for crowded city. Brussels with Carambla succeeds to manage parking spaces without expensive investment to infrastructure, by relying on manual self-services of the owners and the drivers.

We then take a look at the deployment/adjustment cycle. We can see that the effort needed for deployment and adjustment increases as the value of the service becomes larger. It is easy to imagine that having many smart devices and equipment will increases functionality, but decreases the reliability and maintainability. Also as seen in Carambla, maintaining human relationships is also an issue if the service relies on users.

C. Identifying Application of IoT and Self-Aware Systems

We now analyze where and how to apply IoT and self-aware systems to these four services. We consider that the objective of applying the technologies should be either to

- (A1) reduce or assist human tasks, or to
- (A2) improve the performance of system tasks.

For every task identified in Table I, we analyze how the IoT and self-aware technologies can contribute to the task from the viewpoint of (A1) or (A2). In the analysis, we tried to develop as many ideas as possible, based on brain-storming without considering necessary cost or technical details. As a result, 20 ideas have been developed.

Table II summarizes the developed ideas in the same format as Table I. Each idea is labeled by a format in (Ax)[CC-PP-seq], where (Ax) represents the viewpoint, CC identifies a country, PP denotes a process, seq is a sequence number. For example, (A2)[ES-R1-2] identifies the second idea (2) found in the state observation process (R1=1st process of run-time) of Santandar's service (ES=Spain) based on the viewpoint (A2).

First, let us look at the run-time cycle. The state observation using a lot of wireless sensors (in Santander and San Francisco) may cause heavy traffic in limited capacity network. Hence, we may apply self-aware IoT which efficiently transmit the data by collaborating adjacent nodes. Applying self-power-supply IoT contributes to energy reduction as well as long-life battery operation. The manual operations for start/stop parking in Brussels may be automated by a precise location-aware technology. The action proposal process has many rooms to implement *personalized recommendation* based on user's context, preference and finance. In the action execution,

TABLE II. IDEAS OF APPLICATIONS OF IOT AND SELF-AWARE TECHNOLOGIES

Life-Cycle	Process	Grenoble, France	Santander, Spain	San Francisco, USA	Brussels, Belgium
Run-time Cycle	State Observation		(A2) [ES-R1-1] Adopt self-aware IoT for efficient data transmission. (A2) [ES-R1-2] Adopt self-power-supply IoT to reduce energy.	(A2) [US-R1-1] Adopt self-aware IoT for efficient data transmission. (A2) [US-R1-2] Adopt self-power-supply IoT to reduce energy.	(A1) [BE-R1-1] Implement self-aware start/stop parking feature in the app.
	State Interpretation				
	Action Proposal	(A1) [FR-R3-1] Implement location-aware navigation feature in Web or app.	(A1) [ES-R3-1] Implement context / preference-aware parking recommendation feature in app.	(A1) [US-R3-1] Implement context / preference / finance-aware parking recommendation feature in app. (A1) [US-R3-2] Implement location-aware guiding feature in Web or app.	(A1) [BE-R3-1] Implement context / preference / finance-aware parking recommendation feature in app.
	Action Execution	(A1) [FR-R4-1] Implement driving and parking assist feature in the car.	(A1) [ES-R4-1] Implement driving and parking assist feature in the car.	(A1) [US-R4-1] Implement driving and parking assist feature in the car.	(A1) [BE-R4-1] Implement driving and parking assist feature in the car.
Deployment/ Adjustment Cycle	Deployment		(A1) [ES-D1-1] Add self-configuration mechanism to wireless sensors, to allow them automatic connection to the system.	(A1) [US-D1-1] Add self-configuration mechanism to wireless sensors, to allow them automatic connection to the system.	
	Adjustment	(A1) [FR-D2-1] Add self-healing mechanism to toll machine and road sign, to automatically detect failures.	(A1) [ES-D2-1] Add self-healing mechanism to wireless sensors and road signs, to automatically detect failures.	(A1) [ES-D2-1] Add self-healing mechanism to wireless sensors, road signs and smart meters.	(A2) [BE-D2-1] Consider self-protection feature for service platform.

automatic driving and parking assist features would help the driver significantly, although they are quite challenging.

In the deployment/adjustment cycle, the self-aware IoT will help service maintainers and administrators, significantly. In the deployment, sensors with self-configuration mechanism automate the task of connecting the sensors to the system. In the adjustment, applying self-healing mechanism to equipment helps to detect or tolerate faults automatically. In Brussels, it is important to apply self-protection to the service platform, since the platform stores security-sensitive data for the contract.

V. CONCLUSION

In this paper, we proposed the service execution model and the integrated life-cycle model for smart city services. Based on the city-as-a-state-machine concept, the proposed models provides a domain-neutral view of smart city and services. We also investigated four smart parking services using the proposed models. We then developed ideas of where and how IoT and self-aware technologies can be applied.

Adopting the new technologies enriches features of services. However, we must consider carefully that these features are really worth investing or not, based on the value towards the cost and effort of the operation. Otherwise, the service cannot be sustained in the future. Investigation of proper use of technologies and evaluation of sustainability will be left our future work.

ACKNOWLEDGMENT

This research was supported by the Japan Ministry of Education, Science, Sports, and Culture, Grant-in-Aid for Scientific Research (B) (No.26280115, No.15H02701), and CNRS, Projet International de Cooperation Scientifique (PICS06999).

REFERENCES

[1] P. Neirotti, A. D. Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in smart city initiatives: Some stylised facts," *Cities*, vol. 38, no. 0, pp. 25–36, 2014.

[2] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: Towards cooperation frameworks for open innovation," *LNCS*, vol. 6656, pp. 431–446, 2011.

[3] F. H. Priano and C. F. Guerra, "A framework for measuring smart cities," in *15th Annual International Conference on Digital Government Research*, 2014, pp. 44–54.

[4] P. Lalanda, J. A. McCann, and A. Diaconescu, *Autonomic Computing – Principles, Design and Implementation*. Springer, 2013.

[5] L. Gurgen, O. Gunalp, Y. Benazzouz, and M. Gallissot, "Self-aware cyber-physical systems and applications in smart buildings and cities," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2013, pp. 1149 – 1154.

[6] M. Cavada, D. V. Hunt, and C. D. Rogers, "Smart cities: Contradicting definitions and unclear measures," in *Proceedings of the 4th World Sustain. Forum,*, ser. Sciforum Electronic Conference Series, vol. 4, November 2014.

[7] T. Nam and T. A. Padro, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *12th Annual International Conference on Digital Government Research*, 2011, pp. 282–291.

[8] G. Perboli, A. D. Marco, F. Perfetti, and M. Marone, "A new taxonomy of smart city projects," in *17th Meeting of the EURO Working Group on Transportation (EWGT)*, 2014, pp. 2–4.

[9] R. Jucevicius, I. Patasiene, and M. Patasius, "Digital dimension of smart city: critical analysis," in *19th International Scientific Conference on Economics and Management (ICEM)*, 2014, pp. 23–25.

[10] ClouT Project, "ClouT: Cloud of things for empowering the citizen clout in smart cities," http://clout-project.eu/, accessed on 2015-04-03.

[11] IBM Corp., "An architectural blueprint for autonomic computing," in *IBM White paper*, June 2005, pp. 2–4.

[12] S. Hudert, M. Ditze, S. Konig, and V. Fabler, "Transactional service life cycle management in smart electromobility ecosystems," in *IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA)*, September 2012.

[13] VINCI Park, "Leader du stationnement, gere 2,500 parkings dans le monde," http://www.vincipark.com/, accessed on 2015-03-26.

[14] "Smart Santander," http://www.smartsantander.eu/, accessed on 2015-03-26.

[15] Libelium, "Smart city project in santander to monitor parking free slots," http://www.libelium.com/smart_santander_parking_smart_city/.

[16] San Francisco Municipal Transportation Agency, "Sfpark uses innovative technology and advanced pricing strategies to help san francisco," http://sfpark.org/, accessed on 2015-03-26.

[17] Carambla, "Carambla, park smartly," https://carambla.com/, accessed on 2015-03-26.