*Journal of Universal Computer Science, vol. 24, no. 3 (2018), 249-260*
*submitted: 30/9/17, accepted: 5/3/18, appeared: 28/3/18* © *J.UCS*

# A Study on Context-Relationship with Context-Attributes for a Smart Service Generation in Smart City

**Hoon Ko**
(IT Research Institute, Chosun Univ., Gwangju, S. Korea
skoh21@chosun.ac.kr)

**Seogchan Hwang**
(Gensoloft Inc., Seoul, S. Korea
seogchan@gensolsoft.com)

**Libor Mesicek**
(Jan Evangelista Purkyne Univ., Usti nad Labem, Czech Republic
l.mesicek@ujep.cz)

**Jongsun Choi**
(School of Computer Science and Engineering, Soongsil Univ., Seoul, S. Korea
jongsun.choi@ssu.ac.kr)

**Junho Choi**
(Division of Undeclared Majors, Chosun Univ., Gwangju, S. Korea
xdman@chosun.ac.kr)

**Pankoo Kim**
(Department of Computer Engineering, Chosun Univ., Gwangju, S. Korea
pkkim@chosun.ac.kr, Corresponding author)

**Abstract:** This paper is to study how to provide smart services by analyzing each service history based on Cloud computing in smart cities. Normally, users use each smart device to receive their services while they stay in a place. In case they move and visit to other place, the system prepares the service based on their services history. At that time, although the users had visited and had services in past, this visiting may ask other services. Then the existing system usually generates new service, and next it provides the service to the user. However, this method is inefficient, because the system has to accept the number of process whenever all users request the service. In this paper, it suggests the smart service model that it keeps users' services history, which had used them while the user stays to a place. The system catches the user's visiting with the user's contexts in the smart device. The model considers their plan/purpose based on old services, and it decides if they modifies them by editing extra services. It calls the system optimization and the suggesting method shows better efficiency after experimenting.

**Key Words:** Smart Service, Context, Smart City, Cloud, Optimization
**Category:** H.1.1, H.4.2, H.5.5

## 1 Introduction

There are many smart buildings in a smart city, and each building try to provide active and intelligent services to residents and visitors with their smart device while they stay in all places [Bifulco et al., 2016, Nguyen and Jung, 2018]. However, when they generate all services, they don't consider their situations to those services [Gaur et al., 2015, Hong and Jung, 2018]. Also while they receive the services, they may be moving between places. Usually, users use each smart device to have their services available while they stay on some places. In case they may move and visit to other place, the system changes the services based on their services history. Then existing system usually generates new service and provide them to users [Fok et al., 2008, Jung, 2017]. However, this method is not efficient, because the system has to accept the number of process whenever all users request the service. In this paper, the system catches the user's visiting with the user's contexts in the smart device. The model considers their plan/purpose based on old services, and it decides if they modifies them by editing extra services. In this study, it concerns the locations of a smart device, user's movement, and where they had visited with received services [Deakinet al., 2011, Nguyen and Jung, 2017]. That is, it tries to analysis the relationship of the services. To analysis that, it also studies real-space and cyber-space, because users receive all services from smart devices (in cyber-space) in real-space, so it is important to understand two spaces. The smart service model, that we suggests, contains its location catch and context conditions. Finally, the smart service model generates the smart services by using them.

This paper consist of following section, it defines smart services in smart city in section 2, section 3 contains a smart service model, the analysis results are in section 4, and section 5 is focused on conclusion.

## 2 Smart Services in Smart Cities

Smart City consists of a building, residents, and system between building and its residents [Poslad et al., 2015, Lee and Jung, 2017]. IT system contains various IoT devices like vehicles which drive on the high-way, and Users on a public street between buildings or in a place such as home [Pastor et al., 2001, Mance et al., 2010]. The users can move from place to place by vehicles, bikes or on foot with their smart devices. The building where they stay has special characteristics such as its a shape, size (high, width, and wide and so on) [Abbott et al., 2010, Baldauf et al., 2007]. Other buildings have each unique conditions and characteristics. However, all buildings have a common network system, common electronic/gas systems including water system. The systems in the all building operate the same way to residents or to users. Each of them has a different plan and a different purpose why they visit or stay in the area. The applications in the network systems in the all buildings can be based on IoT applications [Yovanof et al., 2009, Bello-Orgaz et al., 2016]. There were no networking systems in past years, so when to transfer a signal to other place, they normally used animals, light, flags or human. On the other hand, these days, we use network system to share or to exchange our information. Also the system could keep all information which they had used, then they can be used for future by following system' or users' requests. Surely to know their location (physical space and cyber space) should be first process to prepare the services [Starns et al.,, 2008]. As it says, they have a smart devices, and they use them to receive services. That is, they are on the street or in a building physically, however they search what they want with the smart devices by using network system, which is a cyber-space. So to provide the information to provide the services to users, that they catch them in cyber-space, it is important to know the physical place where they are [Komninos and Nicos, 2011]. Various applications have been developing with smart devices to use them in our life, for example, an app to control or to share all electronic devices in home or in a building. User can receive all signals via the application while they are away. In the building, it is hard to cover the signal by only one AP, so many APs have dispersed to interconnect them. By interconnecting each other, they communicate the signal, and with the same process they can catch the users' location. The functions with smart devices in a smart city are next;

- The system catches users' plan.
- The system analysis users' future plan.
- The system predicts users' next behaviour and next movement with analysis result.
- The system creates a community of users, who have similar agenda, by understanding users' relationship with each other plans, each future plan, each behaviour and movement.

## 3 Smart Service Model

To design the *smart service model(SSM)*, the system has to trace the contexts [Deakinet al., 2011]. To realize users' plan and users' behaviour on users' location, it can consider all attributes of the context. All moving objects have their attribute such as *time(t)* and *speed(sp)*; also they have their attributes like *location(x, y, h)*, *security(s)* including their *plan (p)* that they input in the smart device [Fig.1]. Usually they receive the services based on this information from the networks while they move following *plan (p)*. However, the servers, networks and smart devices should know all users, attributes, such as their plans or their speed etc, to provide new generating services. The *(SSM)* consists of all *attribute (x, y, h, t, sp, s, p)* to generate new services.

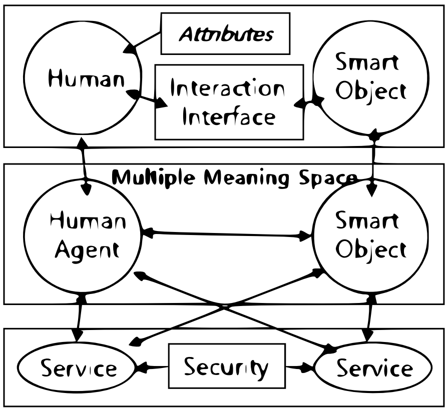


**Figure 1:** Smart Service Flow

### 3.1 Concept

All Services in a smart city have to provide the optimized services to users by using only limited resources. At that same time, they should consider the social relationships between all users who are sharing their contexts such as their plans. However, all users have different plans, even if they are on the same street in a same place/space. For example, there will be a place to work for workers, or to shop for visitors and so on. It means that although they are in/on same place, they can feel differently (one place may have multiple meanings). In addition, all services that are in a space usually should be adapted and evolved from all users' requirements, because their requests will be changed according to social relationships and demands. Therefore, the smart city accepts all demands to offer the best services.

### 3.2 Smart Service Model

There are many people in a space and a place (ex., on a street, in a shopping center, a school and other public buildings); however, they may have why they are there. In this case, each person surely wants to get individualized special services. However, to provide each service to each person is almost not possible. For example, we can assume that some guests visit to a home. Although each person is in the same place, family members will feel comfortable. On the other hand, some guests may feel a little bit strange. In addition, the house is just private family area, but it turns for a special area for guests when they have a meeting to talk about their issues. All meanings of spaces are going to move from a private area to a special area. A Public park can be an another example. all workers think that the park is just the workplace, but all persons who are walking inside will consider that the park is just a place to rest. Therefore, although they are in the same place, some people take it to work and some people can think of it as resting place. Therefore, one space can keep multiple-meanings according to time flow or at the same time. This theory is involved in each service for each person who wants to receive services at their current location. In addition, each service should consider social relationships for each person who shares its place where keeps multiple-meanings and they have to get the optimized services with all resources in a limited space by their requests. *SSM* consists of Service Continuity, Service Relationship, Service Merge, Services Disappear.

• Service Continuity: Generally, users register their plans in a smart device such as an appointment, shopping list and so on by including their visiting place [Niemimaa et al., 2013]. Next it begins to search the old service with $(t_p, p)$ if there are similar services in old service history. It infers the expecting services based on analysis result from old service history which had provided them in terms of plan $(u_{plan})$. Also the new service has to consider the satisfaction of the old service if how much the user was satisfied with the service. This is very important step because it is related with increasing the accuracy of the service and decreasing of the service updating time. It calls *service continuity* and The Fig.2 shows the service continuity.

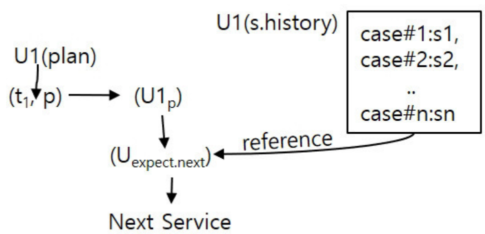


**Figure 2:** Service Sustainability

• Service Relativity: After registering their plan in a smart device, why they are visiting, then they plan to move to other place to fallow their next plans [Niemima and Marko, 2015]. As soon as the user registers his or hers plan, the connected system begins the process to generate the optimized services with their plan and try to offer the services. Even the users, which will visit the same place, they register their plan in case finding a old service which has the same or similar plans. After few seconds or few minutes, it may find exactly matched service or similar service or fully different service with the current plan between when they had visited and when they are visiting. No matter how much they match exactly, similar, or fully different one, just the system provides the services by updating the *users' service* ($S\#_{user}$) based on the information about old services from services history. In addition, they can modify between the old services and the expected services in the system because the services is defined by the contexts that the users registered. Because the system keeps the old contexts, the system can modify and update them which contains *user location*($u_{loc}$), *purpose location*($p_{uloc}$), *current time*($t$), *meeting time*($t_p$), *old service history*($s_{list}$), *user plan*($u_{plan}$), each item define each meaning.

**Algorithm 1** Contexts Structure

```
c#=(u_loc, t, t_p, s_list, u_plan)
u_loc=(u_x, u_y, u_h)
p_loc=(p_x, p_y, p_h)
t=(t_h, t_m, t_s)
t_p=(t_ph, t_pm, t_s)
s_list=(s_1, s_2,..., s_n)
u_plan=(u_p1, u_p2,..., u_pn)

while context ≠ 0 do
    NOTICE_u ← Disappear(S_n)
    CONFIRM M(u)
    KILL(S_n)
end while
```

• Services Merge: The fig. 3 shows the three services to explain how to merge the services. First, in case 1, there are $S_1(c_1,\ c_2,\ c_{3)})$ and $S_2(c_2,\ c_3,\ c_4)$ in old services. Now, the user needs S which has to include $(c_2,\ c_3)$ with $(c_4)$. Therefore, to generate $S$ with $(c_2,\ c_3)$ and $(c_4)$, the system begins to generate s with better evaluated service in $S_1 \cap S_2 = (c_1,\ c_2)$. For example, in case, $S_1$=80 and $S_2$=70, then first it separates $S_1(c_1,\ c_2)$ by removing $(c_3)$ and add $S_2(c_4)$ after removing $(c_3)$, and finally it generates $S_{new(7)}(c_2,\ c_3,\ c_4)$. In case 2, there are also two services; however, there are no common context. Because $S_1$ and $S_2$ have no common contexts, they don't need to compare the result in old services. $S_1$ contains $(c_1,\ c_2,\ c_{3)})$ and $S_3$ has $(c_5,\ c_6)$, new services needs $(c_5)$. Then it generates $S_{new8}$ with $(c_1,\ c_2)$ and $(c_5)$. In case 3, it is full merge algorithm with full used contexts from two services. There are old two services, $S_1(c_1,\ c_2,\ c_{3)})$ and $S_4(c_7,\ c_8)$. And new service wants to use all contexts from two services. Then the system can generate new services named $S_9$ by fully merging $(c_1,\ c_2,\ c_3)$ and $(c_7,\ c_8)$. Therefore, $S_9$ follows next; $S_9(c_1,\ c_2,\ c_3)$.

• Service Separate: There are two service separate cases in this step [Fig. 4]. It tries to optimize the process time by selecting better service and by removing

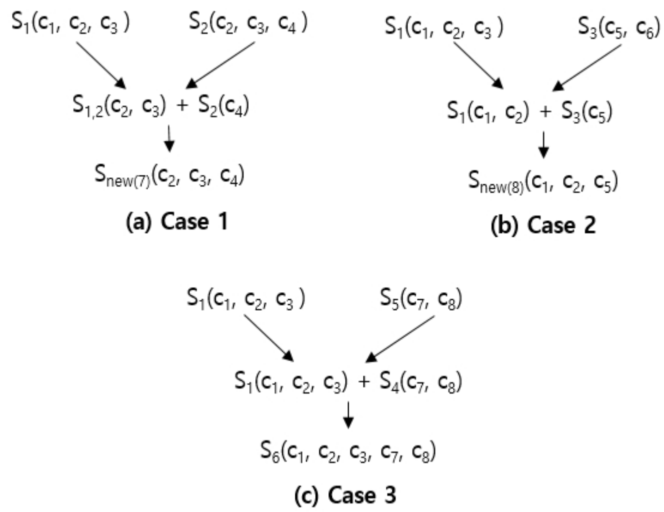


**Figure 3:** A service merge

useless attributes in the listed services [Paluch et al., 2013]. In case 1, it explains the process step when it needs only *(*$c_2$*,* $c_3$*)*. After choosing existing $S_1$*(*$c_1$*,* $c_2$*,* $c_3$*)*, it removes *(*$c_3$*)* from $S_1$, finally it generates $S_8$. As case 2, it would be new service generating step to *context(c)*, the user wants to generate *(*$c_4$*,* $c_5$*)*. However, because there is no service that it has no used, it is necessary to generate $s_{new(9)}$*(*$c_4$*,* $c_5$*)*.

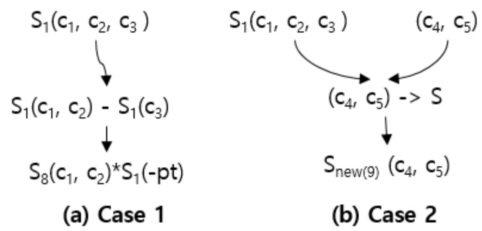


**Figure 4:** A service separate

• Service Disappear: The service in the system will be stored for future use, it updates through service merge or service separate. However, if they don't use over certain time, the S will be disappearing after confirming by the owner.

### 3.3 Smart Service Algorithm

Smart Services(SS) has a similar mission like a cell in human's body. Through its growing by itself, it tries to optimize the service for users, it calls self-growing by repeating generating, disappearing, merging, and separating. In the smart city that has physical space and cyber space at that same time, the smart service will become important. In addition, when it generates or separates the smart services, it has to decide the priority(c) from the contexts first. If it makes up the smart services under FIFO (First In, First Out) from the context in the merge, the user cannot receive the urgent requests to the system in fig.5. Therefore, when it consists of $SM_s$, it has to consider the weight to the contexts. The $wt$ would be evaluated process sequence *(ps)* and service frequency *(sf)*, $wt=ps * sf(_u)$.

```
//context priority decision algorithm
WeightedContext (G, w, s, weight)
    Q <- value
    u := Extract_Min(Q)
    S := S union {u}
        for each v with edge (u, v)*wt;
            if wd[v] > wd[u] + w(u, v)
                wd[v] := wd[u] + w(u, v)
                previous[v] := u
    set wd[v];
```

**Figure 5:** Context Priority Decision Algorithm

There is a merge process between two contexts in smart service definition algorithm. Each context generates various information, so it follows an information theory and has three axiom. It can be defined as *I(s)*, therefore the smart service algorithm follows axiom, *(i)* $I(s) \geq 0$, *(ii)* $I(s_x,s_y)=I(s_x)+I(s_y)$ *which have to be an independent event, and (iii) I(s) is continuous function to the service (s).*

(i) The amount of information is always bigger than 0.

(ii) $I(s_x,s_y)$ follows $I(s^2)=I(s_x)+I(s_y)=2_{xy}I(s)*wt$

(iii) There always exists the amount of information to service *(s)*.

## 4 Analysis

To define the context similarity, it makes the attribute table based on brand, color, price, size from the contexts, and it defines the attribute value's modification to the similar attribute [Table 1]. And it can define the attribute table of each context from the table 1. Table 2 is for the bag's contexts, and the user

**Table 1:** Item Attribute and Value

| Item | Company | Brand | Attribute |
|---|---|---|---|
| Bag, 1 | A, 1 | A, 1 | Black, 1 |
| Bag, 0.8 | A, 0.9 | A, 0.9 | Dark Black, 0.9 |
| Bag, 0.4 | B, 0.8 | B, 0.3 | Black, 1 |
| Bag, 0.5 | B, 0.1 | B, 0.5 | Dark Black, 0.9 |

**Table 2:** Attribute Value

| User | Items($C_1$) | Company($C_2$) | Brand($C_3$) | Prize($C_4$) | Size($C_5$) |
|---|---|---|---|---|---|
| $U_1$, $S_1$ | 0.4 | 1 | 0.3 | 0.6 | 0.8 |
| $U_2$, $S_2$ | 0.7 | 0.6 | 0.7 | 0.9 | 0.3 |
| $U_3$, $S_3$ | 0.3 | 0.8 | 0.2 | 0.8 | 0.8 |
| $U_4$, $S_4$ | 0.1 | 0.4 | 0.1 | 0.5 | 0.8 |

usually requests other services to buy, for example, the table 3 is the service table for *U(Users)*. It is defined in table 3, the users compare the requesting services with the old the services, then it tries to update the service. Finally, it can be the smart service model and it serves. The table 4 shows the merge process between $U_1.S_1$ and $U_2.S_4$. $U_1.S_1$ defines $(C_1, C_2, C_3)$ and $U_2.S_4$ defines $(C_2, C_3, C_5)$. $U_1$ only needs $(C_2, C_3)$ and all service of existing $S_1$. Then, the new service is generated as $S_7$ which has $S_1(C_1, C_2, C_3)+S_4.(C_2, C_3, C_5)$-$S_4(C_5)$. It can define the smart service model by repeating the process. Table 6 shows each amount of information for $C_1$ through $C_5$ after experiment. As we can see the table, $C_2$ has many relation with other $c$, so the amount of information is the highest with 1.9095. And $C_5$ has just 0.6931, also it has no relation with other $c$ but itself. Therefore, according to the result, the service model only have to do is to follow $C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_1 \rightarrow C_5$ when it has to refer the services.

## 5 Conclusions and Future Work

It studied the smart services model to generate the optimized service for users based on their plan inserted in a smart device in this study. The services mix, that the users have to receive for their visit, could be similar with the old services, like, or fully different. The smart service model generates with use of their plan as input; however, it needs some time to create. If there are many users who want to receive their optimized service, the processing time will be increased. To reduce the service generating time to generate new service to user who had visited a place in the past, the system selects the best services, which satisfied

**Table 3:** User Service Table

| Item | Company |
|---|---|
| $U_1$ | $S_1(c_1, c_2, c_3)$ |
| | $S_2(c_2, c_3, c_4)$ |
| $U_2$ | $S_3(c_4, c_5, c_6)$ |
| | $S_4(c_2, c_3, c_5)$ |
| $U_3$ | $S_5(c_7, c_8, c_9)$ |
| | $S_6(c_2, c_4)$ |
| $U_4$ | $S_1(c_1, c_2, c_3)$ |

**Table 4:** Merge with $U_1$ and $U_2$

| $U_1$ and $U_2$ | Items($C_1$) | Company($C_2$) | Brand($C_3$) | Prize($C_4$) | Size($C_5$) |
|---|---|---|---|---|---|
| $C_1$ | 1,- | 0.7,- | 0.8,- | -,- | -,- |
| $C_2$ | 0.5,0.5 | 0.7,0.8 | 0.3,0.1 | -,0.3 | -,0.3 |
| $C_3$ | -,0.8 | -,0.9 | -,0.3 | -,- | -,- |
| $C_4$ | 1.- | 0.3,- | 0.7,- | -,- | -,- |
| $C_5$ | 1,0.9 | 0.7,0.4 | 0.8,0.6 | -,0.5 | -,0.4 |

**Table 5:** Service($S_7$)

| $U_1$ and $U_2$ | Items($C_1$) | Company($C_2$) | Brand($C_3$) | Prize($C_4$) | Size($C_5$) |
|---|---|---|---|---|---|
| $C_1$ | 1.0,1.0 | 0.7,0.0 | 0.8,0.0 | 0.0,0.0 | 0.0,0.0 |
| $C_2$ | 0.5,0.5 | 1.0,1.0 | 0.3,0.1 | 0.0,0.5 | 0.0,0.3 |
| $C_3$ | 0.0,0.8 | 0.0,0.9 | 1.0,1.0 | 0.0,0.0 | 0.0,0.0 |
| $C_4$ | 1.0,0.0 | 0.3,0.0 | 0.7,0.0 | 1.0,1.0 | 0.0,0.0 |
| $C_5$ | 0.0,0.0 | 0.0,0.0 | 0.0,0.0 | 0.0,0.0 | 1.0,1.0 |

**Table 6:** Result($S_7$)

| Context in $S_7$) | Amount of Information |
|---|---|
| $C_1$ | 1.3751 |
| $C_2$ | 1.9095 |
| $C_3$ | 1.5449 |
| $C_4$ | 1.5391 |
| $C_5$ | 0.6931 |

users in the past. Next, it provides the smart services by repeating the new request, adding or deleting useless request in the selected service. As the result, the smart service model could save time of services generation; on the other hand, it has to have prepared a space to store the old services. However, it would be a useful method in terms of providing the optimal intelligent services for visitors. In the future, to reduce the stress, which has to prepare the space to store the services [Bui and Jung, 2018], it needs to study the service integration with the similar services, which were provided to other visitors.

### Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. NRF-2016R1A2B4012638). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2017R1A6A1A03015496).

## References

[Abbott et al., 2010] Abbott, Ken and Freedman, Josh and Newell, Dan and Robarts, Jim (2010). Logging and analyzing context attributes, *Google Patents*, July, 2010.

[Baldauf et al., 2007] Baldauf, Matthias and Dustdar, Schahram and Rosenberg (2007). Florian: A survey on context-aware systems, *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), pages 263-277.

[Bello-Orgaz et al., 2016] Bello-Orgaz, G., Jung, J. J., and Camacho, D. (2016). Social big data: Recent achievements and new challenges, *Information Fusion*, 28, pages 45-59.

[Bui and Jung, 2018] Bui, K.-H. N. and Jung, J. J. (2018). Internet of agents framework for connected vehicles: A case study on distributed traffic control system, *Journal of Parallel and Distributed Computing*, 116, Pages 89-95.

[Bifulco et al., 2016] Bifulco, Francesco and Tregua, Marco and Amitrano, Cristina Caterina and D'Auria, Anna.(2016). ICT and sustainability in smart cities management, *International Journal of Public Sector Management*, 29(2), pages 132-147.

[Deakinet al., 2011] Deakin, Mark and Al Waer, Husam (2011). From intelligent to smart cities, *Intelligent Buildings International*, 3(3), pages 140-152.

[Fok et al., 2008] Fok, Hung Kit and Hui, Chin Ming and Bond, Michael Harris and Matsumoto, David and Yoo, Seung Hee (2008). Integrating personality, context, relationship, and emotion type into a model of display rules, *Journal of research in personality*, 42(1), pages 133-150.

[Gaur et al., 2015] Gaur, A., Scotney, B., Parr, G. and McClean, S. (2015). Smart city architecture and its applications based on IoT, *Procedia Computer Science*, 52, pages 1089-1094.

[Hong and Jung, 2018] Hong, M. and Jung, J. J. (2018). Multi-Sided recommendation based on social tensor factorization, *Information Sciences*, 447, pages 140-156.

[Jung, 2017] Jung, J. E. (2017). Discovering Social Bursts by Using Link Analytics on Large-Scale Social Networks, *Mobile Networks & Applications*, 22(4), pages 625-633.

[Komninos and Nicos, 2011] Komninos, Nicos. (2011). Intelligent cities: Variable geometries of spatial intelligence, *Intelligent Buildings International*, 3(3), pages 172-188.

[Lee and Jung, 2017] Lee, O-Joun and Jung, J. E. (2017). Sequence Clustering-based Automated Rule Generation for Adaptive Complex Event Processing, *Future Generation Computer Systems*, 66, pages 100-109.

[Mance et al., 2010] Mance, Paula and Yu, Peng (2010). Context, relationship transitions and conflict: explaining outcomes for Australian youth from non-intact families, *Journal of Population Research*, 27(2), pages 75-105.

[Niemimaa et al., 2013] Niemimaa, Marko and Jarvelainen, Jonna (2013). IT Service Continuity: Achieving Embeddedness through Planning, Availability, Reliability and Security (ARES), 2013, *Eighth International Conference on*, pages 333-340.

[Niemima and Marko, 2015] Niemimaa, Marko (2015). Extending 'Toolbox' of Business Continuity Approaches, *Towards Practicing Continuity*.

[Nguyen and Jung, 2017] Nguyen, H.L. and Jung, J. E. (2017). Real-time event detection for online behavioral analysis of big social data, *Future Generation Computer Systems*, 66, pages 137-145.

[Nguyen and Jung, 2018] Nguyen, H.L. and Jung, J. E. (2018). SocioScope: A framework for understanding Internet of Social Knowledge, *Future Generation Computer Systems*, 83, pages 358-365.

[Paluch et al., 2013] Paluch, Stefanie and Blut, Markus (2013), Service separation and customer satisfaction: assessing the service separation/customer integration paradox, *Journal of Service Research*, 16(3), pages 415-427.

[Pastor et al., 2001] Pastor, Oscar and Abrahao, Silvia and Fons, Joan (2001). An object-oriented approach to automate web applications development, *International Conference on Electronic Commerce and Web Technologies*, pages 16-28.

[Poslad et al., 2015] Poslad, Stefan and Ma, Athen and Wang, Zhenchen and Mei, Haibo (2015). Using a smart city IoT to incentivise and target shifts in mobility behaviour???Is it a piece of pie, *Sensors*, 15(6), pages 13069-13096.

[Starns et al.,, 2008] Starns, Jeffrey J and Hicks, Jason L. (2008). Context attributes in memory are bound to item information, but not to one another, *Psychonomic Bulletin & Review*, 15(2), pages 309-314.

[Yovanof et al., 2009] Yovanof, Gregory S and Hazapis, George N. (2009). An architectural framework and enabling wireless technologies for digital cities & intelligent urban environments, *Wireless personal communications*, 49(3), pages 445-463.