



# Context-Driven Heterogeneous Interface Selection for Smart City Applications

Inna Sosunova<sup>1(✉)</sup>, Arkady Zaslavsky<sup>1,2</sup>, Alexey Matvienko<sup>3</sup>, Oleg Sadov<sup>1</sup>, Petr Fedchenkov<sup>1</sup>, and Theodoros Anagnostopoulos<sup>1,4</sup>

<sup>1</sup> Department of Infocommunication Technologies, ITMO University, Kronverkskiy pr., 49, St.-Petersburg, Russia  
{inna\_sosunova, pvfedchenkov}@corp.ifmo.ru,  
sadov@linux-ink.ru

<sup>2</sup> Digital Data61, CSIRO, Box 312, Clayton South, VIC 3169, Australia  
Arkady.Zaslavsky@csiro.au

<sup>3</sup> Special Technology Center, Gzhatskaya Street, 21, St.-Petersburg, Russia  
mail@t10000.ru

<sup>4</sup> Research and Education, Ordnance Survey, Southampton SO16 0AS, UK  
Theodoros.Anagnostopoulos@os.uk

**Abstract.** With the diversity and variety of devices and interface modalities these devices offer, the choice of the right interface is still a significant research challenge. We propose a method of Context-driven Heterogeneous Interface Selection for Smart City Applications, which is based on context-driven and situation-aware modality selection mechanism. The method involves the use of a user model, a device model, and an environment model as an adaptation mechanism and a mechanism for selecting an appropriate modality or combination of modalities. Several scenarios of the functioning of the system are described. A series of tests was conducted for each scenario. Tests results are also given in the article. Benefits of the proposed approach are discussed and demonstrated.

**Keywords:** Heterogeneous interface · Knowledge representation  
Context awareness · Smart city · Waste management

## 1 Introduction

Internet of Things (IoT) is enabling Smart Cities and supports data ingestion and digestion [1]. There is not a single universal IoT user interface that can support any possible use case or application scenario. For example, for the driver behind the wheel a voice and/or graphic interface is preferable, for the user in a public place a graphical or textual interface is more suitable. User interface, which provides simultaneously several means of communication: speech, gestures, text, etc., so that the user could choose from useful and convenient tools is called heterogeneous (multimodal) interface [2]. Heterogeneous interface for Smart City applications should also have mechanisms of automatic selection of an appropriate modality or combination of modalities.

In this paper we address the challenge of context-driven and situation-aware modality selection mechanism. We propose a method of Context-driven Heterogeneous Interface Selection for Smart City Applications. The proposed method is based on the concept that the intelligent system must be context-aware and situation-aware. Within the framework of the research, graphics, text, voice, audio, video modalities are conceded.

Validation and implementation of the proposed approach has been carried out as part of Solid Waste Management project, which in turn is a use case of bIoTope project [3]. Efficient waste management is an essential part of Smart City [4], which requires a complex multi-criteria approach [5]. The bIoTope project builds open IoT ecosystems on the basis of innovative IoT technologies and use cases. The project is funded under the EU Horizon 2020 Program. The system described in this paper is part of St. Petersburg pilot of bIoTope project on Smart Waste Management.

The rest of the paper is organized as follows. Section 2 presents Background and Related Work. Section 3 includes description of method of context-driven heterogeneous interface selection. Section 4 describes Implementation of proposed method. Section 5 describes Sample Scenarios and tests carried out at this stage. Section 6 concludes the paper.

## 2 Background and Related Work

### 2.1 Context and Situation

In this paper the context is understood as “any information that can be used to characterize situation of an entity” [6]. We define contextual information as ambient temperature, illumination, noise level, type and characteristics of the user’s device, etc. A context-aware system is a system, which “uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task” [6]. We consider a situation as context abstraction and generalization. The following situations are considered in the article: user is behind the wheel, user walks down the street, user is in a public place, etc.

### 2.2 Related Work

The article [7] describes a heterogeneous interface consisting of a graphical and a voice interface. The task of the interface is to simplify the work of doctors. This system is suitable only for medical specifics and uses only a mobile phone. The article [8] proposes the concept of an interactive system for people with disabilities, equipped with an adaptive context-sensitive user interface. The basic idea is to use a multi-level context processing. The main disadvantage is the absence of implication of the proposed idea. In [9], multi-level data processing is described. The algorithm is based on the use of contextual models and weights. The system can be easily adapted using three context models: user, device and environment. However, the proposed approach assumes that the developer must manually fill in the coefficients. Also, the system is suitable only for a small range of simple scenarios due to its architecture.

### 3 Method of Context-Driven Heterogeneous Interface Selection

Within the Solid Waste Management system, visual and multimedia modalities are necessary for users. Visual modalities include graphics (sending photo, graphical interface) and text; multimedia modalities include audio and video. Each modality has the following uses: input (receiving information from the user) and output (the user receives information from system). Heterogeneous interface can include a single modality or a combination of modalities (complex modality).

We developed a solid waste management (SWM) domain ontology, which is available online [10] for semantic manipulations over entities used in algorithms described further in this paper.

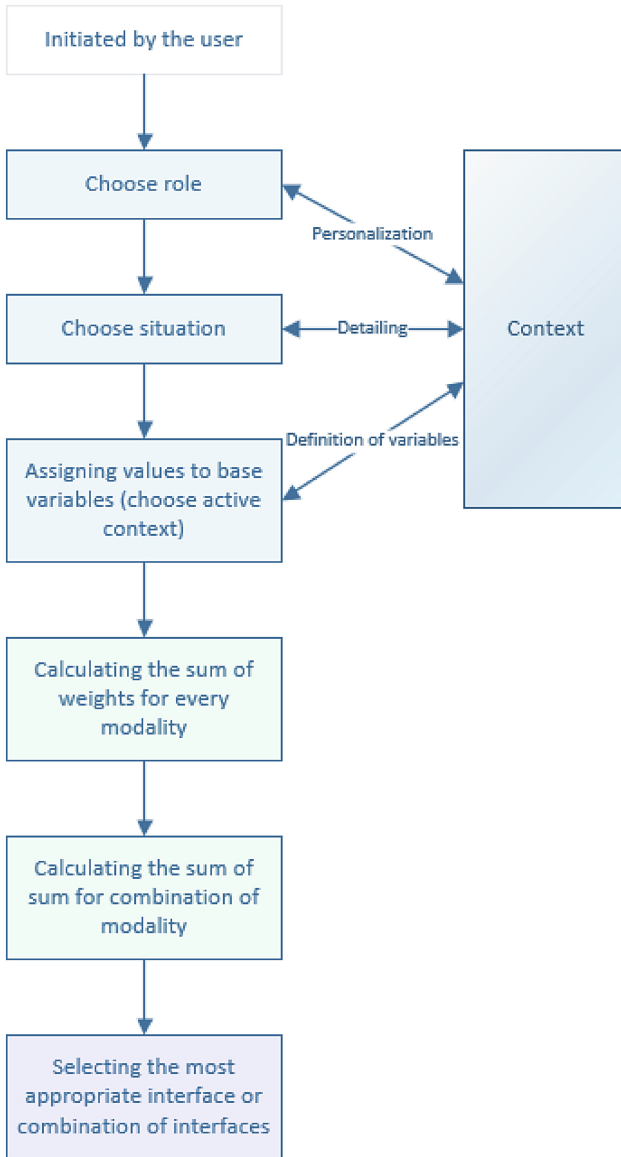
#### 3.1 Appropriate Modality Selection Algorithm

To select a suitable modality or combination of modalities, a multi-step analysis of context is used. The main steps of the algorithm are shown in Fig. 1.

Algorithm contains 4 main parts. At the first stage, we need to personalize user interface. In our case, this means choosing a role. The following user roles were defined: citizen, fleet dispatcher of waste management company, driver, city administration, and janitor. Next step is detailing the situation. Situations determine the environment around the user, which can significantly affect the decision in the choice of interface. Within the framework of the Solid Waste Management use case, 6 main situations were identified:

1. user is in the building
2. user is at home
3. user is behind the wheel
4. user walks down the street
5. user is in a public place
6. user has his hands full.

For each situation, a set of context parameters defining this situation was proposed. To illustrate the algorithm, we use context matrixes. In total, we have three context matrixes (T): the matrix of roles (R), the situation matrix (S) and the base matrix (B) that contains CR, CS and CB context rules respectively.



**Fig. 1.** Appropriate modality selection algorithm

For example, the roles matrix includes rules for define user level like advanced or beginner: *AdvancedIntelegence* and *NoobIntelegence*; situation matrix includes rules *NightEnviroment*, *NoiseEnviroment*, *LightnessEnviroment*. A base matrix is a matrix that contains all context rules that do not contain a role matrix or a situation matrix. For example, *SmarphoneType*, *CPUDevice*.

$$T_m = \begin{bmatrix} w_{11}^m & \cdots & w_{1i}^m \\ \vdots & \ddots & \vdots \\ w_{c1}^m & \cdots & w_{ci}^m \end{bmatrix}$$

Each row of the context matrix ( $T_m$ ) describes how context rule (c) influences modality (i) for selection decision:

$$\text{influence} = \begin{cases} w > 0, & \text{preferred for this context rule} \\ w < 0, & \text{using is undesirable} \\ w = 0, & \text{does not affect} \end{cases}$$

Active context rule is a rule that is activated by context. For example, we could make a rule that works only if the weather is rainy.

In the summation phase, we must select and summarize active context rules (c) for every modality (i) from the context matrices. After that we will calculate the sum of all coefficients for every modality ( $ISUM^i$ ).

$$ISUM^i = \sum_{c=0}^{C^R} R_i + \sum_{c=0}^{C^S} S_i + \sum_{c=0}^{C^B} B_i$$

The third stage represents the sum of coefficients of modalities ( $ISUM^i$ ) according to templates (t). A template is a predefined set of modalities (I) that could be useful together. For example, using text and graphics the best choice for PC in most types of information. Also, we have to separate the types in accordance with the direction (d) of interaction: input and output, because the interaction for reception and transmission can differ among themselves.

$$TSUM_d^t = \sum_{i=0}^{I_d} ISUM^i$$

At the 4th stage, we must select the template that has the maximum sum ( $TSUM_d^t$ ) and modalities (i) in this template (t) will be the best choice for interaction with user ( $W_d$ ).

$$W_d = MAX(TSUM_d)$$

## Pseudocode for appropriate modality selection algorithm:

```

function filterCoefficientsTable(table, activeContext)
    filtredTable = Table()
    foreach contextType in activeContext
        filtredTable.append(contextType, table[contextType])
    endfor
    return filtredTable
endfunction
function calcSumCoef(table, coefficientsUsed)
    fillzeroes(coefSum)
    foreach coefficient in table.keys():
        if (coefficient in coefficientsUsed):
            continue
        coefficientsUsed.add(coefficient)
        foreach modality in modalitySet
            coefSum[modality] += coefficientsTa-
ble[coefficient][modality]
        endforeach
    endforeach
    return coefSum
endfunction
function calcSumTemp(table):
    foreach templateTuple in table:
        foreach modality in templateTuple:
            templateSum[templateTuple] += sumAll[modality]
        endfor
        templateSum[templateTuple] /= length(templateTuple)
    endfor
    return templateSum
endfunction
-----main application-----
Choose role
Choose situation
activeContextTypes = readContext()
roleTable, situationTable, baseTable = readCoefficients()
templateReceiveTable, templateSendTable = readTemplates()
roleTable = filterCoefficientsTable(roleTable, activeCon-
textTypes)
situationTable = filterCoefficientsTable(situationTable, active-
ContextTypes)
roleTable = filterCoefficientsTable(roleTable, activeCon-
textTypes)
fillZeroes(sumAll)
coefficientsUsed = List()
sumAll += calcSumCoef(roleTable[role], coefficientsUsed)
sumAll += calcSumCoef(situationTable[situation], coeffi-
cientsUsed)
sumAll += calcSumCoef(baseTable, coefficientsUsed)
templateReceiveSum = calcSumTemp(templateReceiveTable, sumAll)
templateSendSum = calcSumTemp(templateSendTable, sumAll)
templateReceiveWinner = max(templateReceiveSum)
templateSendWinner = max(templateSendSum)

```

## 4 Implementation

Implementation of the proposed method was carried out on the basis of ITMO SWM project [11]. ITMO SWM project is part of St. Petersburg pilot of bloTope project, which aims to develop The Smart Waste Management System (SWMS). SWMS includes: Central Processing, Web Portal, Mobile applications, communication with IoT (means for collecting and processing data from SGB sensors and sensors installed on waste trucks), the model for IoT data sharing, dynamic route optimization in real time, security and privacy policies, and dynamic route optimization in real time. Web Portal supports three modes of operation: dispatchers of waste collection companies, citizens, and city administration; Mobile applications support two user roles: janitor and driver of waste truck.

The source language for this development is Python. CMS Plone and Zope built-in transactional object database (ZODB) are used for developing web applications. The development framework of Mobile applications is Kivy.

We extended and qualitatively enhanced the functionality of the ITMO SWM described in [11].

Web applications have the following functionality:

- City dashboard (visualization and presentation of current situation with the process of waste collection);
- Various reports on the quality of the collection of solid waste (tools for analyzing the volume and quality of work performed by drivers);
- Driving quality control;
- Traffic jams and road works;
- Complaints & Reviews (mechanism to leave or review a complaint about the quality of waste collection);
- Automated scheduling of vehicles.

Mobile applications have the following functionality:

- display the situation on the map, show the optimal route accordingly the cloud systems data;
- send messages about inability to pick up a garbage bin or other problems encountered, including text, photo, video and voice messages;
- display messages, including text, photo, video and voice messages.

Graphics modality (graphical interface) is represented as (1) City Dashboard, (2) various reports, (3) Traffic jams and road works visualization, (4) Complaints & Reviews. Graphics (sending photo), text and video modalities are implemented using the Messages Module of Mobile application. Audio modality (speech recognition and synthesis) is available for Mobile applications and Web Portal for all users. The implementation of audio modality is based on the algorithm described in [12].

Heterogeneous interface selection module is implemented in Python. The architecture of the developed web application is shown in Fig. 2. Sample scenarios and tests of this system are described in more details in the following section.

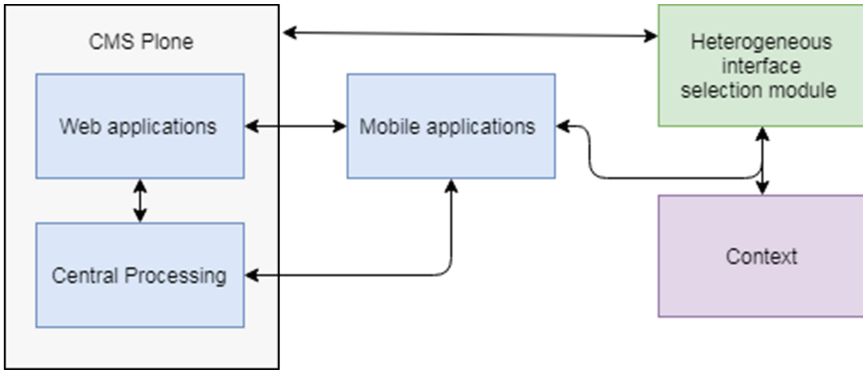


Fig. 2. Architecture

## 5 Sample Scenarios and Testing

Sample Scenario 1. The driver of the waste truck drives at night and requests a route to the garbage bin using smartphone or tablet.

```
-----role: Driver; situation: InCar  
{'receive': (('AudioReceive',), {'Total': 3.0, 'Negative':  
0.0}), 'send': (('AudioSend', 'GraphSend'), {'Total': 2.0,  
'Negative': 0.0})}
```

Fig. 3. Admins interface

The admins interface is shown in Fig. 3.



Fig. 4. Driver's Mobile Application interface on devices Samsung Galaxy S7, Sony Xperia Z3 Tablet Compact и iPad Pro



The driver interface is shown in Fig. 4.

Sample Scenario 2. The citizen is in office (public place) and wants to complain about the quality of garbage collection using PC. The noise level is high.

```

----role: Citizen; situation: InPublic
{'receive': (('TextReceive',), {'Total': 3.0, 'Negative':
0.0}), 'send': (('GraphSend', 'TextSend'), {'Total': 1.5,
'Negative': 0.5})}

```

Fig. 5. Admins interface

The admins interface is shown in Fig. 5.

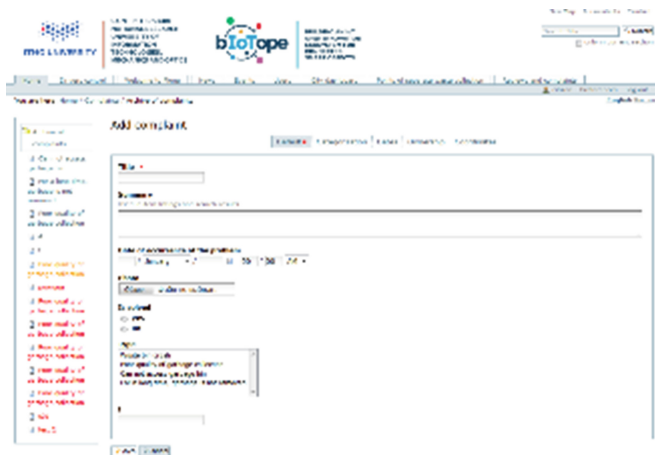


Fig. 6. Citizens interface

The citizens interface is shown in Fig. 6.

## 6 Conclusion

We developed and demonstrated method of Context-driven Heterogeneous Interface Selection for Smart City Applications. It implements automatic selection of an appropriate modality or combination of modalities based on context-driven and situation-aware modality selection mechanism. Several sample scenarios of the functioning of the system are described and tested. It is shown that heterogeneous interfaces allow IoT information system provide better service to its users.

In future work, we will refine the developed method by including larger number of situations and context rules.

**Acknowledgements.** Part of this work has been carried out in the scope of the project bIoTope which is co-funded by the European Commission under Horizon-2020 program, contract number H2020-ICT-2015/ 688203 – bIoTope. The research has been carried out with the financial support of the Ministry of Education and Science of the Russian Federation under grant agreement RFMEFI58716X0031.

## References

1. Zanella, A., Bui, N., Castellani, A.V.L.: Internet of Things for smart cities. *Internet Things J.* **1**(2), 22–32 (2014)
2. Karpov, A.A.: Assistive information technologies based on audio-visual speech interfaces. In: SPIIRAS Proceedings, vol. 27, pp. 114–128 (2013)
3. The bIoTope Project Webpage [Electronic resource]. <http://www.biotope-project.eu/>
4. Sakai, S., Yoshida, H., Hirai, Y.: International comparative study of 3R and waste management policy developments. *J. Mater. Cycles Waste Manag.* **13**, 86–102 (2011)
5. Anagnostopoulos, T., et al.: Challenges and opportunities of waste management in IoT-enabled smart cities: a survey. *IEEE Trans. Sustain. Comput.* **2**(3), 275–289 (2017)
6. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. *Comput. Syst.* **40**(3), 304–307 (1999)
7. Alnanih, R., Ormandjieva, O., Radhakrishnan, T.: Context-based and rule-based adaptation of mobile user interfaces in mHealth. *Procedia Comput. Sci.* **21**, 390–397 (2013)
8. Zouhaier, L., Yousra, B.H., Ben Ayed, L.J.: Building adaptive accessible context-aware for user interface tailored to disable users. In: *Proceedings of International Computer Software and Applications Conference*, pp. 157–162 (2013)
9. Macik, M.: *Automatic User Interface Generation Doctoral Thesis*, 173 p (2016)
10. SWM-DomainOntology [Electronic resource]. <http://sdn.naulinux.ru:8128/SPB/swm-ontology.owl>
11. Sosunova, I., Zaslavsky, A., Anagnostopoulos, T., Fedchenkov, P., Sadov, O., Medvedev, A.: SWM-PnR: ontology-based context-driven knowledge representation for IoT-enabled waste management. In: Galinina, O., Andreev, S., Balandin, S., Koucheryavy, Y. (eds.) *NEW2AN/ruSMART/NSCC -2017. LNCS*, vol. 10531, pp. 151–162. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-67380-6\\_14](https://doi.org/10.1007/978-3-319-67380-6_14)
12. Sosunova, I., Zaslavsky, A., Anagnostopoulos, T., Medvedev, A., Khoruzhnikov, S., Grudin, V.: Ontology-based voice annotation of data streams in vehicles. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) *ruSMART 2015. LNCS*, vol. 9247, pp. 152–162. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-23126-6\\_14](https://doi.org/10.1007/978-3-319-23126-6_14)