UDC 519.711:004.896:004.047 DOI: 10.18698/0236-3933-2020-4-4-21

LEVERAGING CONTEXT AWARENESS IN DESIGNING MOBILE E-GOVERNMENT

E. Agbozo eagbozo@urfu.ru
A.N. Medvedev a.n.medvedev@urfu.ru

Ural Federal University, Ekaterinburg, Russian Federation

Abstract

In the age of ubiquitous computing, smart systems, internet of things, and numerous modern technological advances in the smartphone world, context-aware programming has made progress within the past decade. Google's Awareness API is a great example of the available power to developers for optimizing user experience of mobile applications. Electronic Government (e-government) solutions, primarily mobile-based, which stand to gain maximum utility should integrate such innovations. This study, supported by the PMJ (Perception–Memory–Judgment) cognitive computing model, integrates context-aware models into e-government design in order to increase e-participation and user experience with respect to public service delivery. The study has employed the ontological evaluation technique which is a recommended non-empirical method of evaluating information systems. The purpose of using this technique is validating potency of the model in the real-world. The study conceptualizes the model and verifies it by the non-empirical technique of ontological evaluation using Protégé. This study simulates a scenario using C# and illustrates the feasibility of the model by considering user-privacy and system health

Keywords

Context-aware, e-government, ontological evaluation, PMJ model, cognitive computing, user experience

Received 13.05.2020
Accepted 20.07.2020

Introduction. E-government systems are of great benefit to society all over the world. They play a vital role in the digital society by providing users (citizens and residents of a country) by easier and convenient access to public services through the medium of information technology (IT). With the increase of mobile technology adoption especially in the developing economies, there have been several mobile e-government (m-government) solutions across the globe to take its advantage in better public service delivery [1]. Researches have revealed that out of over 40 e-government implementation projects in developing or transitional countries, 35 % of them were total failures, 50 % were partial failures, and only 15 % were successful [2]. Also in the United States, over 3 trillion dollars were estimated to be spent on information technology projects by governments between 2000 and 2010; making up for an overall estimated failure rate of 60 % [3]. Their study discussed the issue of design–reality gap with respect to the required performance for the potential number of users.

Failure of e-government has been linked to numerous reasons such as bad design and design-reality gaps. Electronic participation (e-participation) is a necessary metric to assess the adoption rate of e-government implementations. An increase in e-government adoption, which is a sign of positive e-participation, gives an indication of e-government projects success [4]. Therefore; user needs must be factored into the design of e-government.

This work proposes the need for integrating context-awareness into e-government mobile applications. The main purpose of this research is curbing the rate of failure of e-government and increase e-participation, primarily in the developing economies.

Context-aware approach to design. Context-aware systems sense or remember information about the person and the emotional or physical situation in order to reduce computer-user communications and efforts [5]. With the emergence of pervasive and ubiquitous computing, context-awareness has become a very important component which must be considered in the design of smart systems [6]. The authors summarized context-aware models requirements as follows: ease of development; heterogeneity and mobility; machine-interpretable representation; quality of context; contextual information partitioning; evolvability, extensibility, flexibility, and applicability; comparability; traceability; timestamping; satisfiability; relationships and dependencies; usability of modeling formalisms; and efficient context provisioning. Thus, modeling context-awareness into the previously mentioned concepts must be considered in order to attain optimal contextual-awareness in any system. Advantages of context-awareness modeling into systems include: improving user experience, predicting future contexts, and increasing system efficiency [7].

The research in [8] recommended context-aware design for building safety critical context-aware systems in automobiles. The purpose was to understand, predict and improve driver behavior. The study in [9] also focused on context-driven recommendation systems. Another study applied context-awareness to model a novel means of including location (gyroscope and accelerometer) and supervised learning in authenticating smartphone users as a security layer [10].

A study showed that there exists a level of scarcity regarding the application of context-aware models into web service-based systems [11]. However; it's value is clear today in internet of things (IoT)/internet of everything (IoE) for obtaining flexibility and adaptability to changes in network behavior [6, 12].

One of the most prominent context-aware integration in mobile technology is Google's Awareness API. Its goal is to build assistive and aware applications which assist users as well as provide recommendations. Google's desire to build a great experience for users is evident in their approach to incorporate context-aware solutions into the smartphone experience. Google's Awareness API combines the following seven contexts into one API solution: Local time; Device latitude and longitude; Place; User activity; Nearby Eddystone beacons; when and whether or not headphones are plugged in; and Local weather conditions [13]. These options give developers the opportunity to build great mobile solutions on top in order to attain maximum utility and user experience. Google's Awareness API has two APIs within — Fence API which reacts to changes in the user's environment and the Snapshot API which obtains instant details about the user's current environment by accessing 7 signals from one simple API surface [14].

Researchers in [15] developed vStore, a context-aware framework for mobile micro-storage at the edge. It consists of a context aggregator that makes use of Google's Awareness API for identifying the user's current activity (i.e., whether they are still driving or walking). To meet the need of integrating data and services provided by various applications, another study proposed a model for an on-the-fly integration of applications using context-aware self-adaptation approach for application service [16].

Public sector e-services have not integrated such solutions into their design approaches. E-government mobile applications can benefit from such initiatives in order to increase user participation and interaction with government services. The need for more context-aware models in the current digital society is highly necessary as user needs vary increasingly. Therefore; a cognitive computing model must be the backbone of context-aware applications. The main question of this study is: How can context-aware models be integrated into e-government mobile experiences?

To answer this research question, the methodology for deriving the desired end-product has been represented in the research workflow shown in Fig. 1. The subsequent sections delve deeply into how each procedure of the workflow was undertaken.

Proposed context-aware model for mobile e-government solutions. E-government is a multi-agent system with multiple stakeholders and complexity. Thus; there is a necessity for integrating context-awareness into e-government design. The purpose of this study is to solidify the proposed model's efficiency. Therefore; it is theoretically based on the PMJ (Perception–Memory–Judgment)

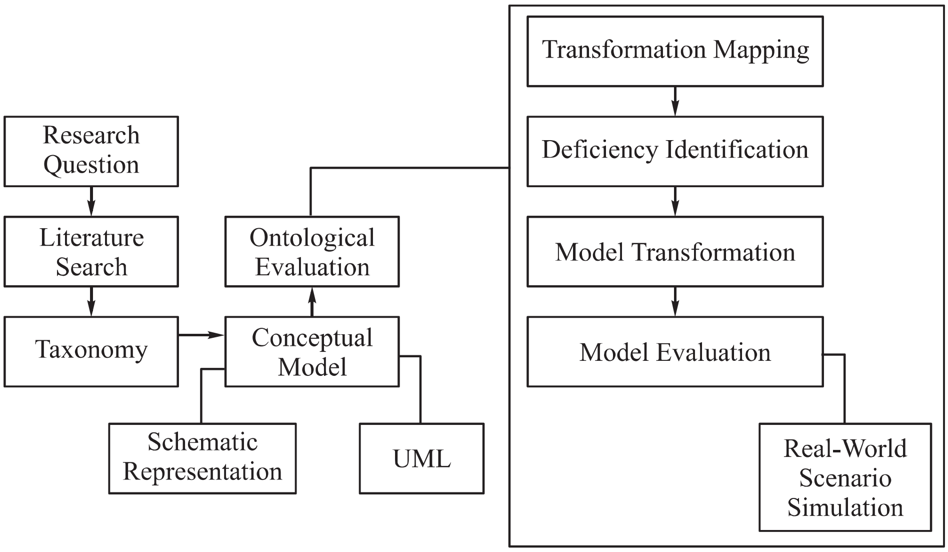

Fig. 1. Research workflow (source: own elaboration)

model from a cognitive computing perspective. The work formalizes the context-aware model in mobile e-government (m-government) systems based on the PMJ model.

The PMJ model shown in Fig. 2 is built upon human cognition. It is made up of three stages and pathways that integrate cognition and computability [17]. In the stage of perception, salient features are extracted. During the memory stage, encoding and storage processes occur. During the judgment stage, efficient decisions are made through categorization learning and encoding. As shown in Fig. 2, the case of a cognitive computing model — precisely the PMJ model — in context-aware modeling of e-government services is depicted as follows:

a) when an event occurs such as a vehicular accident or traffic jam where a user is in the vicinity or in close proximity, or a user finding itself geographically close to a public service location. In this case the application notifies the user. Also, an event could also be triggered by another user or group of users who will notify the necessary public service office regarding any impending danger. This is followed by sending a callback to users in close proximity;

b) users have the prerogative to respond to the e-government application recommendation in order to pay a fine they owe, escape traffic jams, set a reminder to make payments for certain services, or be aware of important information.

Thus; the main constructs of the PMJ model perfectly satisfy the proposed context-aware constituent of m-government as a multi-agent system. Fig. 3 illustrates a case for context-aware modeling in m-government.

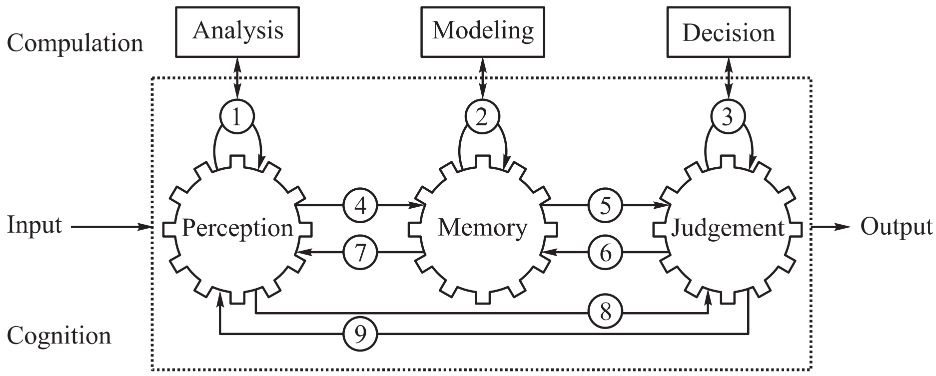

Fig. 2. A schematic representation of the PMJ model (source: [17])

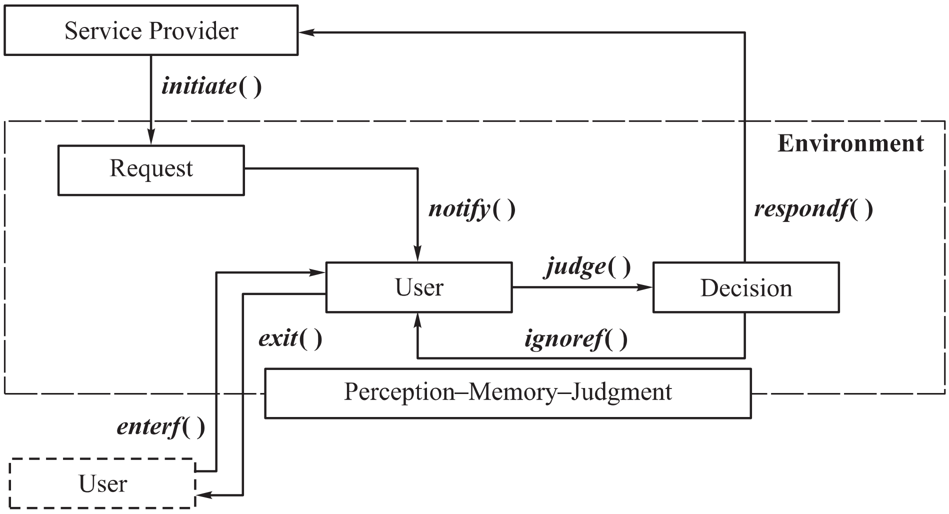

Fig. 3. A schematic representation of the formalized context-aware model of mobile e-government solutions (source: own elaboration)

In Fig. 3, the proposed model is schematically represented. Service providers (i.e., public service providers or government agencies) notify users when they enter a certain environment (i.e., an event or they are within the radius of the service provider) — "*enter () function*". Therefore; the environment establishes the context for which service providers can make requests and users can respond. The user perceives the request, evaluates and makes a judgment whether to respond or ignore. At this point, the PMJ cognitive computing model is in play. It should be noticed that the advantage of this model is that, non-users who may find themselves within the environment are not interacted with. The goal is to preserve user privacy and smartphone health by giving citizens control over what they allow. The user makes judgment before responding to the provider request and makes the transaction or accesses the service.

Fig. 4 illustrates the proposed use-case of context-aware e-government interaction represented as a sequence diagram in UML (Unified Modeling Language) format. It highlights two states of user interaction (i.e., user_in and user_out) with respect to interaction with service provider's requests within a context-aware e-government setting. The user enters the environment and interacts with services contextually such as G2C (government-to-citizen) services. It must be noticed that, users are in control of accepting or rejecting requests. It is clear from Fig. 4 that at each instance, users are capable of exiting the context-aware environment.

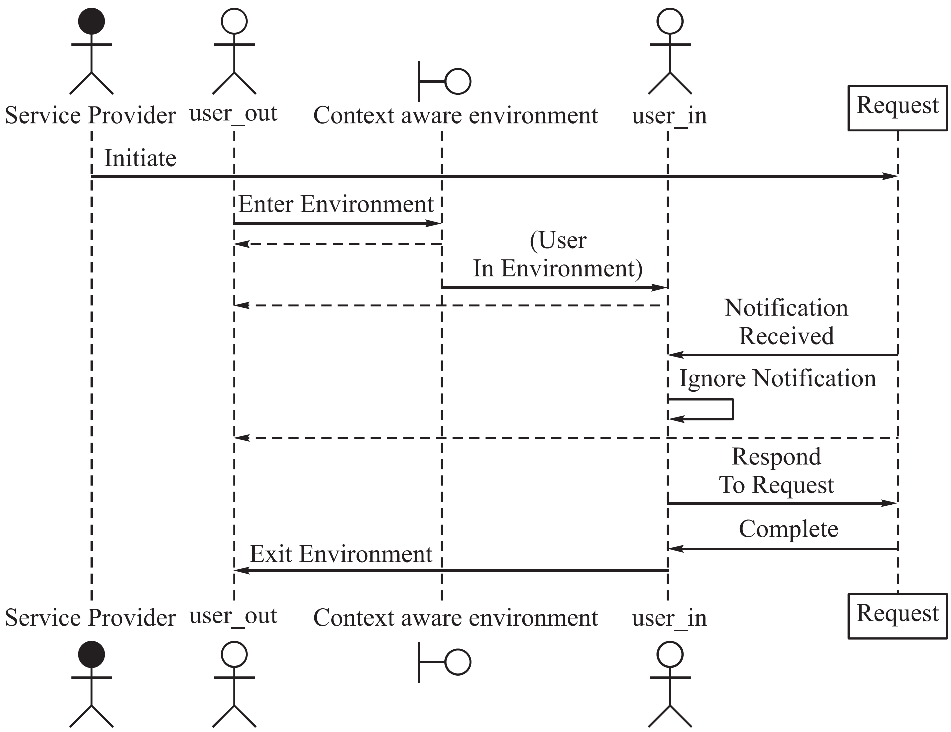

Fig. 4. UML sequence diagram representation of context-aware e-government interaction (source: own elaboration)

Conceptual model validation. When proposing any contextual model, it is necessary to provide evaluation approaches. Despite its importance, studies have pointed out that in information systems (IS) research, evaluation is significantly challenging due to the complex nature of both the object of investigation and evaluation itself since both aspects are subject to individual paradigmatic viewpoints [18]. This study adopted ontological evaluation as a non-empirical technique for evaluating the proposed context-aware mobile e-government concept. The reason is non-existence of empirical data at the time of doing the research. The non-empirical technique of ontological evaluation of IS which was recommended by researchers in [19] is suitable for this study. Ontology-based techniques are meant for systematic evaluation of a number of domain representation artifacts. These artifacts include reference information models, simulation frameworks and many other IS which are capable of significantly help in ensuring that these artifacts preserve the model's intrinsic logical as well as its ontological consistency [20]. The ontological evaluation technique was chosen as the preferred approach so as to prevent unwarranted expenditure by government authorities since the modeling approach is capable of combining real-world scenarios with the necessary constraints and conditions. This gives designers a full picture of what is required for obtaining a working final solution. Hence, for this research contribution, the model has been evaluated by its provision of syntactic, semantic, and aesthetic rules expressed in the algorithmic expressions and simulation phases. Therefore; it serves as a framework for embedding a context-aware services architecture into mobile e-government services. It provides end users (i.e., e-government stakeholders) with quality public service delivery as well as proactive governance in the modern digital society.

In performing an ontological evaluation, the following steps are followed:

1) transformation mapping development;

2) ontological deficiencies identification;

3) model transformation;

4) results assessment [21].

In order to achieve an efficient evaluation, Protégé software has been utilized for each of the four steps. For step 1, mapping the modeling constructs is represented in Table. Where the context-aware model's conceptual constructs are mapped to ontological constructs. The World Wide Web Consortium (W3C) Ontology Web Language (OWL) has also been adopted for representing ontologies. The reason is that it has been used in previous research for similar purposes — specifically, in an ontological representation of cybercrime [22].

Mapping the Business Capability Conceptual Model Constructs to Ontological Constructs

Context-aware model of mobile e-government solutions	
UML construct (from sequence diagram in Fig. 4)	Ontological construct (OWL2 DL)
Actor	Class
Object	Class
Attribute	Object property
Messages	Object property
Boundary	Class

Upon successfully representing the model's taxonomies in an ontological structure using the mapping guide from Table, steps 2 (adopting the classifications of deficiencies used in [21]), 3 and 4 are subsequently performed in the Protégé[*] platform. The ontological representation clearly shows reality of the conceptual model in identifying deficiencies. Furthermore it fits the legal requirements towards a formal ontology for transforming UML into ontological constructs [23]. Thus, it is represented as follows:

1. Class (*owl:Thing*): *Stakeholder, Service Provider, Individual, User* and *Non-User*

Where Service Provider $\wedge$ *Individual* $\in$ *Stakeholder. Therefore; every entity is a stakeholder in the e-government ecosystem.*

Where: {Non-User, User} $\in$ *Individual and Non-User* $\cap$ *User* = $\emptyset$.

Users and non-users are disjoint entities despite the fact that they are all stakeholders. This also means, that a non-user is capable of being a user.

2. Object Property (*owl:topObjectProperty*): *EnterEnvironment, ExitEnvironment, Ignore, Respond,* and *SendRequest*

In order to express the ontologies in a real-world case scenario, Individuals and Data Properties are also embedded into the ontological model. Hence, a test-driven approach is applied with the following constructs:

3. Data Property (*topDataProperty*): *e-service, mobileDevice, personType, taxNumber,* and *userID.*

4. Individuals: *GovDept1, GovDept2, person1, person2, person3* and *person4.*

i. Where: *{ GovDept1, GovDept2}* $\in$ *ServiceProvider*

a. *GovtDept1 = {e-service: "Social Security Contribution", e-service: "Foreign Business Yearly Levy", e-service: "Local Business Yearly Levy"*}

b. Where, *GovtDept2 = {e-service: "Traffic Fine", e-service: "Police Report Collection"}*

ii. Where: *{person3, person4}* $\subseteq$ *Non-User* $\wedge$ *{person1, person2}* $\subseteq$ *User*

a. *person1 = {personType: "Citizen", userID: "govApp_0004", mobileDevice: "Android", taxNumber: "13897"}*

b. *person2 = {personType: "Foreign Resident", userID: "govApp_1309", mobileDevice: "iOS", taxNumber: "91238"}*

c. *person3 = {personType: "Citizen", mobileDevice: "iOS", taxNumber: "12345"}*

d. *person 4 = { personType: "Foreign Resident", mobileDevice: "Android"}*

[*] Protégé — Stanford University. Available at: https://protege.stanford.edu (accessed: 14.07.2020).

The class hierarchy has been constructed using Protégé (open-source ontology editor and framework for building intelligent systems) and illustrated in Fig. 5. The figure depicts the previously discussed relationships among class elements of the model. It should be noticed that though a non-user is a stakeholder, it cannot respond to service providers' requests within the context-aware environment. Therefore; all these constraints have been taken into consideration during the ontological evaluation step-wise procedure.

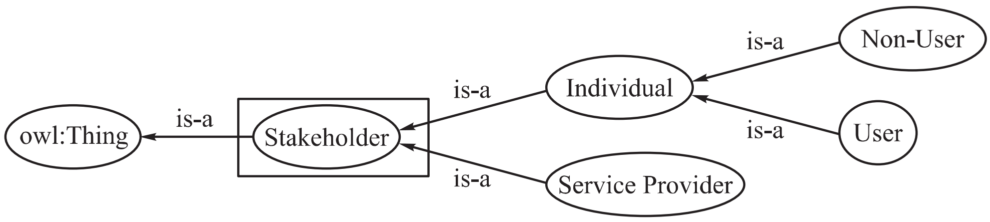

Fig. 5. Class hierarchy of ontological representation of a context-aware e-government model in Protégé (OWL Viz)

Fig. 6 and 7 have been visualized using OntoGraph and ComodIDE plugins of Protégé framework respectively [24, 25]. They illustrate the finalized OWL ontological representation including all constraints and components building blocks. It is evident that non-users (*person3* & *person4*) have no relationship with service providers (*GovDept1* & *GovDept2*). The reason is that both are citizens and foreign residents with no userID. The conceptual model's constructs have been perfectly verified by the ontological axioms and constructs. Finally, the model has been evaluated by simulating (coding) one scenario or more as recommended by the researchers in [26].

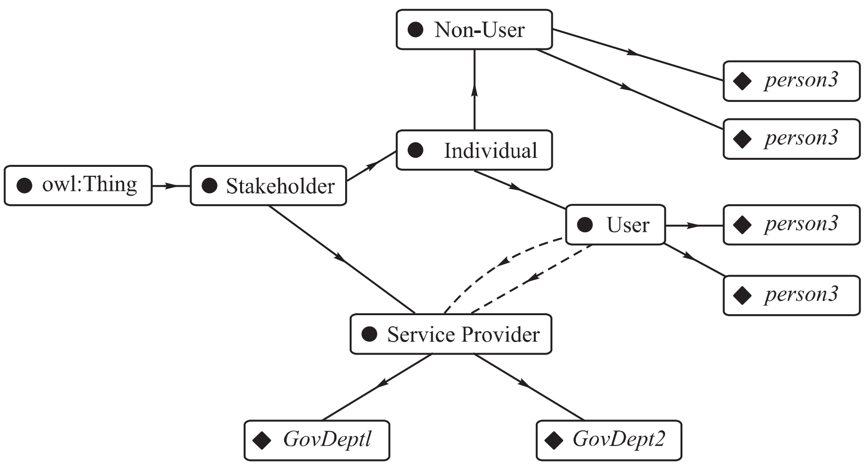

Fig. 6. OntoGraf visualization of ontological representation of a context-aware e-government model in Protégé

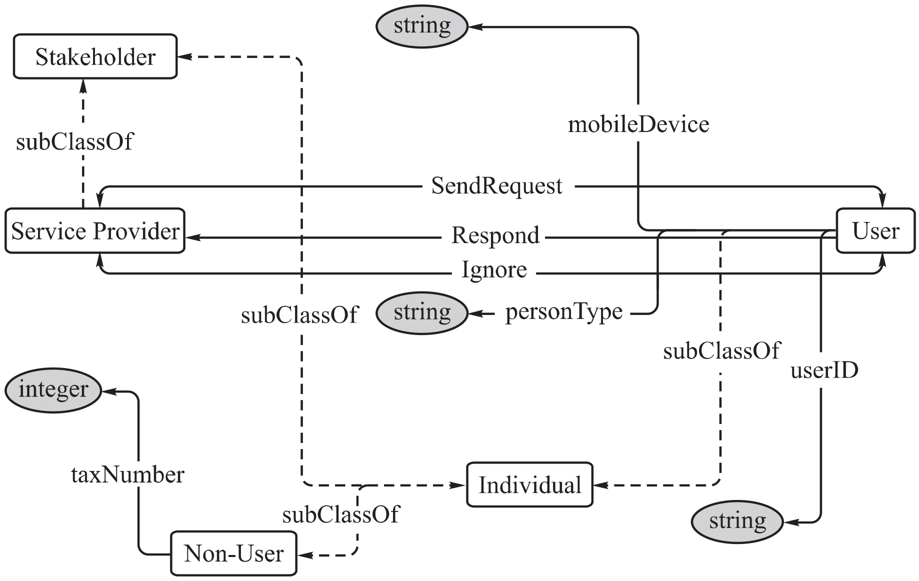

Fig. 7. ComodIDE visualization of ontological schema of a context-aware e-government model in Protégé

In order to illustrate the possibility of integrating the PMJ cognitive computing model into context-aware design of e-government mobile solutions, a simulation has been performed by modeling the conceptual model in C# programming language using Microsoft Visual Studio. This simulation has been followed by coding the functions and debugging various instances of users and non-users interactions with a context-aware environment.

Fig. 8 contains the functions that make up the context-aware model components using C# simulation. The *notify()* function — broadcasts notifications of available services to potential users. The *ignore()* function — controls the user's action to ignore notification requests. The *pmj()* function — handles the perception-memory-judgment cognitive computing aspect of the model where user makes decisions whether to respond or deny a request notification. The *initiate()* function — broadcasts signals from the provider and verifies whether users are within the vicinity or not. The *enter()* function — controls the user's choice to enter the environment or not. The *respond()* function — activates the user's ability to respond to a notification request or ignore completely while being in the environment. The *reject()* function — discards provider request giving the user autonomy.

Fig. 8 and 9 show the functions to be performed in the context-aware environment with respect to the interaction between e-government applications and end users. Simulation of the model includes coding and debugging with three instances.

```
public static void notify()
{
    exit();
    Console.WriteLine("\n Notification Request -[ Accept (1) / Reject (0)]");
}
public static void ignore()
{
    Console.WriteLine("\n Ignore User");
    Console.ReadKey();
}
public static void initiate()
{
    Console.WriteLine("User Verfication: App User or Not?");
    user = int.Parse(Console.ReadLine());
}
public static void enter()
{
    Console.WriteLine("\nEnter Environment or Not?");
    person_state = int.Parse(Console.ReadLine());
}
public static void pmj()
{
    request_state = int.Parse(Console.ReadLine());
}

public static void respond()
{
    Console.WriteLine("\n Notification Accepted - Perform Action?  (1) / Ignore  (0)");
    transact = int.Parse(Console.ReadLine());
    exit();
}
public static void reject()
{
    Console.WriteLine("\n Notification Rejected");
    exit();
}

 public static void exit()
 {
     Console.ForegroundColor = ConsoleColor.Red;
     Console.WriteLine("Do you want to exit environment?");
     Console.ForegroundColor = ConsoleColor.White;
     int x = 0;
     x = int.Parse(Console.ReadLine());

     if (x == 1)
     {
         Console.WriteLine("....User Has Exited Environment....");
         Console.ReadKey();
         System.Environment.Exit(1);
     }

 }
```

Fig. 8. Simulation of context-aware model as C# functions

In Fig. 10, the console reveals simulation results obtained from the interaction between the user and the requests made by the service provider. User requests to exit the environment are made also due to the fact that the user may want to exit the environment at any time in the real-world scenario.

```
static void Main(string[] args)
{
    //Provider INITIATE function
    initiate();

    //USERs only are eligible
    if (user == 0) {
        Console.WriteLine("\nIgnore Individual");
        Console.ReadKey();
    }
    else
    {
        Console.WriteLine("\n ----- User Mode Activated-----");
       //ENTER ENVIRONMENT function- person_state
        enter();
        if(person_state == 0){
           //user IGNORED
            ignore();
        }
        else{

            //NOTIFY function
            notify();
            //PERCEIVE - JUDGE - MEMORY function
            pmj();

            if(request_state == 0){
               //REJECT request function
                reject();
            }
            else{

                //RESPOND to request function
                respond();
                if(transact==1){
                    Console.WriteLine("\n Action Completed");
                    Console.ReadKey();
                }else{
                    Console.WriteLine("\n Exit");
                    Console.ReadKey();
                    //If user wants to EXIT ENVIRONMENT
                    exit();
                }
            }
        }
    }
}
```

Fig. 9. Simulation of context-aware model — main method

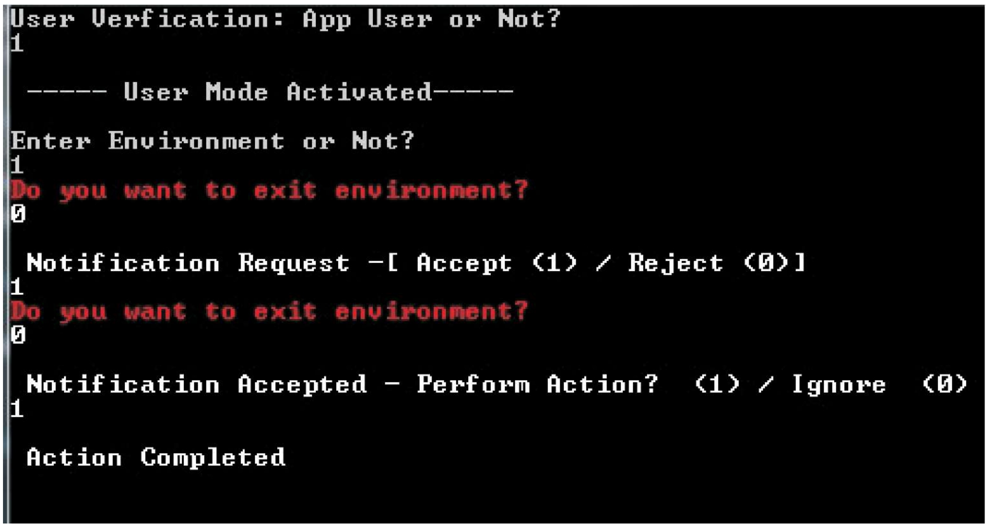

Fig. 10. Simulation of context-aware model — test result for user that reacts with the provider(s) requests in the environment

Fig. 11 illustrates the simulation results made by provider's evaluation of whether a random individual is a user or not. This is feasible with the existence of location-based services switched-on on the user's smartphone.

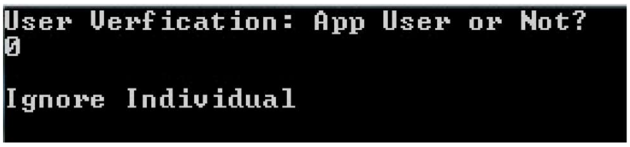

Fig. 11. Simulation of context-aware model — test result for non-users

Fig. 12 depicts a scenario where a user does not interact with the environment, thus it is ignored by the service provider. Therefore; the model is aware of user-privacy as well as smartphone health by not bothering the user.

Fig. 13 provides the last test case scenario where a user enters the environment, and does not interact with the service provider's notification. Thus, the user has full autonomy over its actions.

The above described model has the potential of bringing to the forefront the possibilities with respect to e-government within a mobile context in certain environments based on location of users. One example is the prevention of road traffic to all users within a perimeter. Another example is notifying users of debts owed to public sector departments when they are within close proximity (since humans tend to be forgetful). Another use case that embodies the model is renewal notifications of vehicle insurance or drivers' licenses when drivers are within the vicinity of the public service offices. Finally, the context aware model with respect to e-government is applicable within services that may notify users to evaluate government initiatives when users are within close proximity. An example could be evaluating and verifying road or pavement reconstruction, traffic light or street light repairs, or e-government self-service terminals repairs when the user is close to the object of interest.

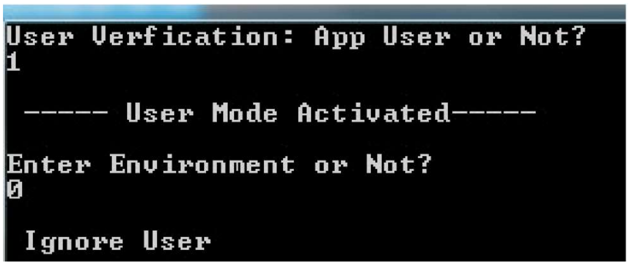

Fig. 12. Simulation of context-aware model — test result for a user that does not react with the environment

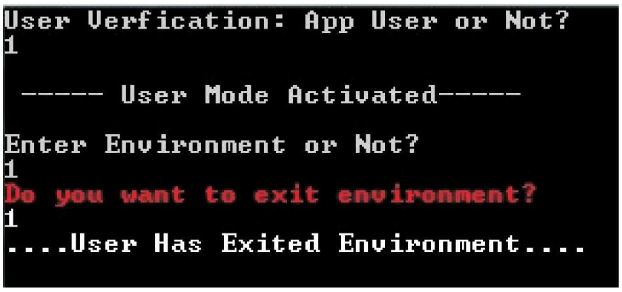

Fig. 13. Simulation of context-aware model — test result for a user that enters the environment and exits without interaction with the system notification

Conclusion. Amidst the numerous opportunities for context-aware modeling in e-government services are the issues of user-privacy, ethical implications, and system health. These issues are of concern to e-participation. In light of the benefits of context-aware integration into e-government, the repercussions must also be curbed by the model. These repercussions can be discussed as follows:

a) the issue of ethics and privacy is to be addressed by enforcing user autonomy and control (permission required) which are activated and deactivated by users. Context-aware applications are untrusted and considered to be adversarial. The reason is although they are authorized to collect users' data, they are perceived to extract users' private information [27]. The specification of privacy design guidelines and privacy preservation strategies must be embedded into the design procedure. Wang and Zhang recommended privacy preservation techniques which are more context-based rather than the generic location-based privacy approaches [27]. Legislative requirements and provisions must be enforced in order to protect users, thereby leading to human-centric e-government systems. Transparency within context-aware models are attainable by integrating efficient, human-centric and privacy-assured requirements into context-aware e-government applications;

b) system health must also be considered in order to reduce consuming the end users' smartphone battery or pressure on memory leading to sluggish devices or data communication. According to the conducted studies, the best energy-preserving algorithm is the one that infuses into the low-level sensory operations by manipulating the frequentness of sensory sampling intervals. The purpose is to address power efficiency in context awareness and preserve accuracy as well as effectiveness [28]. Therefore, in context-aware design of e-government applications, integration of optimization engines, infrastructure, and algorithms that collect performance measures is recommended. The aim is to increase energy efficiency, and reduce network congestion while preserving user experience. System health can preserved by resolving the latency, accuracy, and battery (LAB) abstraction recommended by previous researches. This tension can be resolved by means of the Senergy API which increases both programmer productivity and energy efficiency [29].

It must also be noticed that standardization of government design processes leading to efficient interoperability, easy sharing of data across agencies and departments, and building semantic web systems for the Web 3.0 must be considered by service providers (i.e., public sector departments). According to past studies, it is recommended that context modeling must be sensitive enough to distinguish different program contexts. The purpose is to prevent unnecessarily annoying prompts to the user. These prompts may be a result of a slight difference in the program execution and which is treated as a new context [30].

Conclusion. This work has presented a novel approach for automatically recommending public e-services (i.e., e-government) to users based on context. The conceptual model incorporates context-awareness models in e-government systems. In order to increase electronic participation (e-participation) in e-government, it is important for government (service providers) as stakeholders, to integrate novel means such as context-aware models into the design of e-government. User privacy is an important element to be considered when building context-aware systems. User privacy has been taken account in this work. The model has been furthermore validated using non-empirical method of ontological evaluation. This has been done on the basis of its provision of syntactic, semantic, and aesthetic rules expressed in the algorithmic expressions and simulation of the conceptual model.

With respect to theoretical contributions, this research contributes to the context-aware models literature into e-government and its design. Also, the work theoretically contributes to the application of ontological evaluation as a non-empirical technique in verifying conceptual models. Regarding practical contributions, this research presents pragmatic steps into improving e-government design. Another practical contribution of this work is the application of Protégé and W3C's OWL in ontologically modeling and evaluating e-government processes to prevent future contradictions during implementation of projects of such magnitude. Hence, supporting return on investment as a result of taking all necessary conditions and scenarios into consideration is time wasting.

Due to some limitations of the study such as the lack of empirical evaluation, it is recommended that future studies extend on the model, its simulation, and the development of metrics to evaluate the performance of context-aware design approaches in e-government. Moreover; future works could explore gamification of context-aware e-government models so as to involve users with rewards while aiming to e-government development which leads to increased quality of service provision.

Translated by Authors

REFERENCES

[1] Faisal N., Talib F. E-government to m-government: a study in a developing economy. *IJMC*, 2016, vol. 14, no. 6, pp. 568–592.
DOI: https://doi.org/10.1504/IJMC.2016.079301

[2] Meiyanti R., Utomo B., Sensuse D.I., et al. E-government challenges in developing countries: a literature review. *CITSM*, 2018.
DOI: https://doi.org/10.1109/CITSM.2018.8674245

[3] Lessa L. Sustainability framework for e-government success: feasibility assessment. *Proc. ICEGOV2019*, 2019, pp. 231–239. DOI: https://doi.org/10.1145/3326365.3326396

[4] Bennett D. Factors influencing the success of an E-participation project in South Africa. PhD Thesis. University of Cape Town, 2015.

[5] Selker T., Burleson W. Context-aware design and interaction in computer systems. *IBM Syst. J.*, 2000, vol. 39, no. 3-4, pp. 880–891.
DOI: https://doi.org/10.1147/sj.393.0880

[6] Pradeep P., Krishnamoorthy S. The MOM of context-aware systems: a survey. *Comput. Commun.*, 2019, vol. 137, pp. 44–69.
DOI: https://doi.org/10.1016/j.comcom.2019.02.002

[7] Shafaei S., Müller F., Salzmann T., et al. Context prediction architectures in next generation of intelligent cars. *ITSC*, 2018, pp. 2923–2930.
DOI: https://doi.org/10.1109/ITSC.2018.8569617

[8] Rakotonirainy A. Design of context-aware systems for vehicle using complex systems paradigms. *Proc. CONTEXT-05 Workshop on Safety and Context*, 2005.
Available at: https://eprints.qut.edu.au/2700 (accessed: 18.05.2020).

[9] Irfan R., Bickler G., Khan S.U., et al. Survey on social networking services. *IET Netw.*, 2013, vol. 2, no. 4, pp. 224–234.

[10] Primo A., Phoha V.V., Kumar R., et al. Context-aware active authentication using smartphone accelerometer measurements. *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2014, pp. 98–105. DOI: https://doi.org/10.1109/CVPRW.2014.20

[11] Truong H., Dustdar S. A survey on context-aware web service systems. *Int. J. Web Inform. Syst.*, 2009, vol. 5, no. 1, pp. 5–31.
DOI: https://doi.org/10.1108/17440080910947295

[12] Mongiello M., di Noia T., Nocera F., et al. Context-aware design of reflective middleware in the internet of everything. *STAF*, 2016, pp. 423–435.
DOI: https://doi.org/10.1007/978-3-319-50230-4_33

[13] Vigliarolo B. Google awareness API. Ehe smart person's guide. TechRepublic, 2017.

[14] Google Awareness API. *developers.google.com: website.*
Available at: https://developers.google.com/awareness (accessed: 13.05.2020).

[15] Gedeon J., Himmelmann N., Felka P., et al. vStore: a context-aware framework for mobile micro-storage at the edge. *MobiCASE*, 2018, pp. 165–182.
DOI: https://doi.org/10.1007/978-3-319-90740-6_10

[16] Sinha A., Mehta N., Bhakta P., et al. On the fly integration of applications using context aware approach. *IJARIIT*, 2019, vol. 5, no. 3, pp. 435–438.
Available at: https://www.ijariit.com/manuscripts/v5i3/V5I3-1302.pdf

[17] Fu X., Cai L.H., Liu Y., et al. A computational cognition model of perception, memory, and judgment. *Sc. China Inf. Sc.*, 2014, vol. 57, no. 3.
DOI: https://doi.org/10.1007/s11432-013-4911-9

[18] Recker J. Conceptual model evaluation: towards more paradigmatic rigor. *EMMSAD'05: Tenth International Workshop on Exploring Modeling Methods in Systems Analysis and Design.* University of Porto, Portugal, pp. 569–580.

[19] Derguech W. Business capability-centric management of services and process models. PhD Thesis. Galway, National University of Ireland, 2014.

[20] Barcelos P.P.F., Guizzardi G., Garcia A.S., et al. Ontological evaluation of the ITU-T Recommendation G.805. *18th Int. Conf. Telecommunications*, 2011, pp. 232–237. DOI: https://doi.org/10.1109/CTS.2011.5898926

[21] Aileen C.-S., Latif A.-H. Information systems research methods, epistemology, and applications. IGI Global, 2008.

[22] Barn R., Barn B. An ontological representation of a taxonomy for cybercrime. *Proc. ECIS*, 2016. Available at: https://aisel.aisnet.org/ecis2016_rp/45 (accessed: 18.05.2020).

[23] Evermann J. A UML and OWL description of Bunge's upper-level ontology model. *Softw. Syst. Model*, 2009, vol. 8, no. 2, pp. 235–249.
DOI: https://doi.org/10.1007/s10270-008-0082-3

[24] Falconer S. OntoGraf. Protégé Wiki, 2010.

[25] Shimizu C., Hammar K. CoModIDE — the comprehensive modular ontology engineering IDE. *ISWC*, 2019, vol. 2456, pp. 249–252.

[26] Noy N.F., McGuinness D.L. Ontology development 101: a guide to creating your first ontology. Available at: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf (accessed: 18.05.2020).

[27] Wang W., Zhang Q. Privacy preservation for context sensing on smartphone. *IEEE ACM Trans. Netw.*, 2016, vol. 24, no. 6, pp. 3235–3247.
DOI: https://doi.org/10.1109/TNET.2015.2512301

[28] Yürür Ö., Liu C.H., Perera Ch., et al. Energy-efficient and context-aware smartphone sensor employment. *IEEE Trans. Veh. Technol*, 2015, vol. 64, no. 9, pp. 4230–4244. DOI: https://doi.org/10.1109/TVT.2014.2364619

[29] Kansal A., Saponas S., Bernheim Brush A.J., et al. The latency, accuracy, and battery (LAB) abstraction: programmer productivity and energy efficiency for continuous mobile context sensing. *ACM SIGPLAN Notices*, 2013, vol. 48, no. 10, pp. 661–676.
DOI: https://doi.org/10.1145/2509136.2509541

[30] Zhang M., Yin H. Efficient, context-aware privacy leakage confinement for Android applications without firmware modding. *Proc. ASIA CCS*, 2014, pp. 259–270.
DOI: https://doi.org/10.1145/2590296.2590312

Agbozo E. — Post-Graduate Student, Senior Lecturer, Department of Big Data Analytics and Video Analysis Methods, Ural Federal University (Mira ul. 19, Ekaterinburg, 620002 Russian Federation).

Medvedev A.N. — Cand. Sc. (Eng.), Assoc. Professor, Department of Big Data Analytics and Video Analysis Methods, Ural Federal University (Mira ul. 19, Ekaterinburg, 620002 Russian Federation).

Please cite this article as:

Agbozo E., Medvedev A.N. Leveraging context awareness in designing mobile e-government. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2020, no. 4, pp. 4–21. DOI: https://doi.org/10.18698/0236-3933-2020-4-4-21