A Context Server to Allow Peripheral Interaction

Borja Gamecho, Luis Gardeazabal and Julio Abascal

Egokituz: Laboratory of HCI for Special Needs, University of the Basque Country (UPV/EHU), Donostia, Spain

{borja.gamecho, luis.gardeazabal, julio.abascal}@ehu.com

Abstract. This paper presents a research to create a mobile context server application that provides other applications with complex context information. The main objective is to avoid disrupting or overwhelming users with explicit requests for data that can be obtained otherwise by the interpretation of combined sensor data. It is mainly aimed at mobile devices used by people with disabilities to allow them to interact with local services supplied by means of ubiquitous computing.
Keywords: Context awareness, people with disabilities, accessible ubiquitous computing.

1 Introduction

There is an increasing variety of services provided by local machines, such as ATMs, information kiosks, vending machines, etc. These services are frequently inaccessible for people with disabilities because they are equipped with rigid user interfaces. Nevertheless, the application of Ubiquitous Computing techniques allows access to intelligent machines through wireless networks by means of mobile devices. Smartphones can provide an excellent way to interact with ubiquitous services that would otherwise be inaccessible. People with disabilities can benefit from this type of interaction if they are provided with accessible mobile devices that are well adapted to their characteristics and needs.

The INREDIS[1] project created a ubiquitous computing environment to allow people with disabilities to interact with locally provided services. In this project our laboratory developed EGOKI [1], an automatic interface generator that is able to create adapted and accessible user interfaces that are downloaded to the user device when she or he wants to access a ubiquitous service.

Nevertheless, when users are immersed in an "intelligent environment" they can become overwhelmed by the quantity of explicit interactions that they have to manage through their mobile device. For this reason we are working on ways to enhance a mobile device's context awareness to ease the interaction with the aforementioned services.

[1] http://www.inredis.es/default.aspx

2 Related Work

User attention is an interesting concern for interaction with a ubiquitous system. The work of Weiser & Brown (1997) distinguishes two levels of attention: central and peripheral. The central attention focuses on the main task that is being addressed by the user, while the peripheral attention is related to "what we are attuned to without attending to explicitly" [2]. Additionally, in multitasking environments the user’s attention can be negatively affected by interruptions. Leiva et al. (2012) reported that interruptions while interacting with an application can delay by up to four times the completion of a task in a mobile environment [3]. Thus, two conclusions can be drawn: it is desirable to ensure that users can pay attention to applications around them without feeling overwhelmed; and it should be attempted to maximize the focus of the user on a single central task, reducing shifting between tasks.

Two different ways to get the attention of the user are described in the following lines. On the one hand, it is possible to merge background interaction with peripherical attention. The work of Bakker et al. (2012) [4] presents an interactive system called FireFlies to explore the way in which primary teachers are able to manage secondary tasks in the periphery of their attention. The intention is to study “how interaction with technology can fluently blend into people’s everyday routines, similar to the way in which interactions with the physical world are a part of routines”. Using this approach, tasks that would require direct attention or a cognitive effort disappear from the central attention of the users. On the other hand, a slightly different approach is to consider the implicit human-computer interaction. Schmidt (2000) [5] defined this as “An action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input.”. This work studies different sources to add implicit information, with the most relevant to this work being “sensing context using sensors”. Schmidt described sensor-based perception as a way to recognize the implicit context and illustrates some examples that can help to manage interruptions and limit the need for input when users are interacting with computers. Therefore, the implicit context can be useful to free a user’s attention from a specific task.

Concerning the supporting technology, two approaches stand out in the literature: The first frees the user's attention by using wearable devices. Saponas (2010) defines the always-available interaction, describing methods to interact with a mobile device without using it explicitly [6]. Likewise, a user can receive notifications from applications using smartwatches as a second screen at a glance[2]. The second proposal enhances the context-awareness of ubiquitous applications using smartphones. Smartphones and the sensors within them are useful to characterize activities and recognize context information. Reddy et al. (2010) were able to distinguish between the movements of the smartphone user (stationary, walking, running, biking, and travelling in a motor vehicle outside) using the GPS receiver and the accelerometer [7]. In a similar way, the work of Wiese et al. (2013) recognizes whether a mobile is in a bag, in a pocket or in the hand [8].

[2] Sony Smartwatch (http://www.sony.com/SmartWatch) or Pebble (http://getpebble.com/)

3 A Context Server for Peripheral Interaction

In our case, peripheral interaction includes all the implicit activities that are conducted to interact with an application. Our objective is therefore to collect, by means of sensors, any type of information that helps the device to manage the interaction, thus minimizing the need for explicit user participation. This is called context information and we gather it by means of sensors that are located in the mobile device, either worn by the user or deployed in the environment.

Usually, each mobile application has to collect and process data from the sensors available in the device in order to adapt the interaction to the context. This is frequently done in real time and competing with other applications, which limits the possibilities to extract complex results.

Our approach focuses on a context server application that collects data from the sensors, combines this, and extracts complex information that can be directly used by the other applications.

3.1 From Sensing to Perception

In order to determine what information is provided in each case, we created a sensor taxonomy that classifies the different types of sensors that are currently found in mobile devices or worn by users. This taxonomy allows us to work with “abstract sensors” independently of their specific datasets.

To extract combined information we developed an ontology of sensors, including rules that specify the type of information that can be obtained from the combination of different sets of sensors.

3.2 From Perception to Interaction

Our context server can contribute to peripheral interaction by providing the applications with valuable information that would otherwise be explicitly requested from the user.

The context provider can assist developers to make use of the context in a simpler way. For instance, the context server allows applications to select the most appropriate modality to interact with a user with communication restrictions, due to disability or to a situational impairment. For instance, if the microphone detects that the local level of noise is too high the application can avoid voice commands and prioritize text or images; or, if the inertial sensors detect that the user is walking, driving or riding a bicycle, touch input can be switched to voice input.

In addition, some applications for people with disabilities use the server to perform their tasks without disturbing the user. In the following lines four examples of freeing users’ attention using our context server application approach are described.

3.2.1 Affective Interaction

Affective computing focuses on detecting and reacting to emotions by using computers. Emotional information can be useful to understand and detect the context of the user when interacting with an application. The work of Haag et al. (2004) presents an example of inferring a user's mood and emotions using physiological signals [9] obtained via sensor devices that measure heart rate variation, perspiration, respiration rate, skin temperature, etc.

The context server application can detect and manage the data from the wearable sensor devices and infer information to feed applications with information about the mood of the user. This is valuable for the peripheral interaction. For instance, it is possible to avoid stressful situations that occur when a user has to attend to too many tasks simultaneously. In a similar way automatic rearranging of the tasks can be performed to distinguish the enjoyable ones from the annoying ones.

3.2.2 Smart Wheelchair

Smart wheelchairs are robotic platforms to assist people with mobility restrictions to navigate the physical environment. Smart wheelchairs are equipped with sensors (sonar, laser range finder, bump sensors, etc.) in order to perceive elements that can affect the navigation. Thereby, diverse modes of operation are developed to assist the user including: collision avoidance, wall following or close approach to objects [10].

Controlling a smart wheelchair with a joystick can become a stressful task. Situations such as approaching a narrow space or going through a door may require a high level of concentration. In such a scenario, the context server application would discover and integrate the wheelchair sensors. The data collected is helpful to infer when the user is facing a stressful situation. The context application server provides this information to the wheelchair, which can trigger automatic guidance procedures.

3.2.3 Smart Traffic Lights

There are smart traffic lights that assist people with special needs to cross the street safely. For instance, current Audible Pedestrian Signals[3] (APS) attached to traffic lights help people with vision impairments to know when they can walk across a pedestrian crossing. In addition, works such as UCARE [11] present prototypes for scenarios where impaired users can negotiate via their mobile devices the period required to cross the road. If the user has to handle the device when approaching a pedestrian crossing, his/her attention is disrupted. However, this task is moved to the periphery by using the context application server. The speed and position of the user are gathered using accelerometers and GPS and sent to the traffic lights to activate the APS. Moreover, the mobile device can negotiate, in the background, the time required to cross without the explicit participation of the user.

[3] APS are also called accessible pedestrian signals: http://www.apsguide.org/index.cfm

3.2.4 Peripheral Interaction with EGOKI

As mentioned in the introduction, EGOKI is a UI generator for ubiquitous services. The user's abilities, device characteristics and service functionalities are taken into account to create an accessible UI. For each function of the service or application, EGOKI selects the appropriate input/output elements to ensure a suitable interaction [1].

The context server application empowers EGOKI to allow peripheral interaction for some applications. Firstly, it helps to detect appropriate input and output methods; for instance, by allowing the use of gestures when a wearable device with accelerators or an electromyogram is detected. Secondly, it helps to choose the communication modality in an accurate way. For instance, when blind users are in a noisy environment, avoiding speech and audible channels would be an issue. Instead of that, the volume of the user device should be adapted to the noise level. Finally, when the context application server provides accurate information about the user, the UI generation process avoids having to explicitly ask the user for that information. For instance, when an application needs the user location it is provided by the context server and EGOKI excludes that input element from the final UI.

Therefore, some ubiquitous services will not require explicit attention from the user and due to the change of modality would run in the periphery of the user's attention.

4 Discussion and Conclusion

There are a number of issues that require our attention when merging peripheral interaction with the context server application.

To begin with, peripheral interaction can be contradictory with a well established practice of activity-aware systems. As mentioned by Mahmud et al. (2009) [12], activity-aware systems must inform the users to correct failures in activity recognition to avoid mistakes and manage uncertainty. This would increase the number of interactions that a user must perform and would consequently draw his/her attention more than necessary.

In addition, context information depends on the set of sensors detected by the context application server. Users can be affected by the loss of smartness when the availability of sensors changes. This is related to the "masked uneven conditioning" challenge stated by Satyanarayanan (2001) [13].

Moreover, the application domain is a key factor for activity recognition. The accuracy of the activity and emotion recognition techniques "in the field" frequently produces worse results than in the laboratory. In a similar way, the accuracy of the results depends on the person.

Finally, the impact on a user's privacy must be considered, because large quantities of data about the user are collected and logged. These data must be protected to avoid their unauthorized use; for instance, by commercial applications.

The combination of sensor data allows the interpretation of the context at a higher level, providing mobile applications with implicit methods of interaction that augment communication without disrupting the user's attention for routine adjustments.

Acknowledgments

EGOKITUZ is funded by the Department of Education, Universities and Research of the Basque Government (grant IT395-10). Borja Gamecho holds a PhD scholarship from the Research Staff Training Programme of the Basque Government.

References

1. Abascal, J., Aizpurua, A., Cearreta, I. et al.: Automatically Generating Tailored Accessible User Interfaces for Ubiquitous Services. In: 13th international ACM SIGACCESS conference on Computers and accessibility, pp. 187--194. ACM, New York (2011)
2. Weiser, M., Brown, J. S.: The coming age of calm technology. In: Denning P.J., Metcalfe R.M. (eds.) Beyond calculation, pp. 75--85. Springer, New York (1997)
3. Leiva, L., Böhmer, M., Gehring, S., Krüger, A.: Back to the app: the costs of mobile application interruptions. In: 14th international conference on Human-computer interaction with mobile devices and services, pp. 291--294. ACM, New York (2012)
4. Bakker, S., van den Hoven, E., Eggen, B.: FireFlies: supporting primary school teachers through open-ended interaction design. In 24th Australian Computer-Human Interaction Conference, pp. 26--29. ACM, New York (2012)
5. Schmidt, A.: Implicit human computer interaction through context. Personal and Ubiquitous Computing, 4(2-3), 191--199. Springer-Verlag (2000)
6. Saponas, T. S., Adviser-Landay, J. A.: Supporting everyday activities through always-available mobile computing. University of Washington, Seattle (2010)
7. Reddy, S., Mun, M., Burke, J., Estrin D., Hansen M., Srivastava M.: Using Mobile Phones to Determine Transportation Modes. J. *ACM Trans. Sen. Netw.* 6, 2, Article 13 (2010)
8. Wiese, J., Saponas, T. S., Brush, A.: Phoneprioception: Enabling Mobile Phones to Infer Where they are Kept. In: SIGCHI Conference on Human Factors in Computing Systems 2157--2166. ACM, New York (2013)
9. Haag, A., Goronzy, S., Schaich, P., Williams, J.: Emotion recognition using bio-sensors: First steps towards an automatic system. In: André, E., Dybkjær, L., Minker, W., Heisterkamp, P. (eds.) Affective Dialogue Systems. LNCS, vol. 3068, pp. 36--48. Springer, Berlin Heidelberg (2004)
10. Simpson, R. C.: Smart wheelchairs: A literature review. *J. Rehabil. Res. Dev.*, *42*(4), 423--436 (2005)
11. Vales-Alonso, J., Egea-López, E., Muoz-Gea, J. P., García-Haro, J., Belzunce-Arcos, F., Esparza-García, M. A., Gonzalez-Castao, F. J.: Ucare: Context-aware services for disabled users in urban environments. In: The Second *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008,* pp. 197--205. IEEE Press (2008)
12. Mahmud, N., Vermeulen, J., Luyten, K., Coninx, K.: The five commandments of activity-aware ubiquitous computing applications. In: Duffy, V.G. (eds.) Digital Human Modeling. LNCS, vol. 5620, pp. 257--264. Springer, Berlin Heidelberg (2009)
13. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Comm., 8(4), pp. 10--17. IEEE press (2001)