International Journal of Computer Applications Technology and Research
Volume 3– Issue 1, 1 - 4, 2013, ISSN: 2319–8656

Software Engineering Challenges in Pervasive Computing: A review

Surabhi Soni
Engineer,
Mahindra Comviva,
Bangalore, India

Ayush Rastogi
Development Engineer,
Laurus Infosystems,
Bangalore, India

Abstract-Moving away from decades of machine-centric computing and making pervasive human-centric computing, the new wave of computing, a reality revolutionizes the relationship between humans and computing systems. There is a growing interest in the use of context-awareness as a technique for developing pervasive computing applications that are flexible, adaptable, and capable of acting autonomously on behalf of users The software challenges to turn such pervasive or ubiquitous computing environments into reality are enormous In this paper, we review some of the challenges of software engineering in pervasive computing.

Keywords: Human-centric computing, Pervasive computing, Software Engineering, Ubiquitous computing, human-machine interaction.

1. INTRODUCTION

Pervasive computing introduces a radically new set of design challenges when compared with traditional desktop computing. In particular, pervasive computing demands applications that are capable of Operating in highly dynamic environments and placing minimal demands on user attention. Late Michael Dertouzos, pointed out the need for pervasive computing. He stated the following:

If computers are to live up to the promise of serving us, they will have to change drastically and never again subject us to the frustrating experiences we all have shared [Dertouzos, M.L.2001][1].

It is envisioned that pervasive computing systems will help people to achieve more while doing less. These systems will:

- Understand us when we speak to them;
- Do much of our routine brainwork for us;
- Get us the information we want, when and where we want it;
- Help us work with other people across space and time;
- Adapt on their own to our individual needs and desires [Dertouzos, M.L.2001] [1].

In the pervasive computing era, we will not need to carry our own physical devices with us anymore. Instead, configurable generic devices, either handheld or embedded in the environment, will bring computation to us, whenever we need it and wherever we might be. As we interact with these *anonymous* devices, they will adopt our information personalities. They will respect our desires for privacy and security.

Pervasive human-centric computing systems will change how businesses, organizations and governments work with each other, as well as how individuals interact. It represents the dawn of a new era in Information Technology (IT) [Dertouzos, M.L.2001][1].

To shift the focus of computing from machines to humans, major changes are required not only in technologies and systems, but also in the approach to developing, deploying and managing technologies and systems. Weiser presented his vision for pervasive human centric computing in 1991. He further articulated his vision as follows:

There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the hum an environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods [Weiser, M.1993] [Weiser, M.1991][3][4].

2. MOTIVATION

Challenges which are based on natural characteristics of pervasive computing systems (i.e. mobility, dynamism and heterogeneity) can be evaluated from a more domain specific perspective,

Hence, we particularly list following basic interrelated requirements for such pervasive learning environments:

(1) *Device independence*: applications and data should be always accessible without any device dependence,

(2)*Application independence*: data should be always accessible without any application dependence,

(3)*Adaptivity and Adaptability*: learning environment and elements of this environment should dynamically adapt according to context of learner(s) and users should be able to configure such environments such as composing/decomposing data and applications,

(4)*Collective Operation*: applications in such environments must be able to collectively operate for the benefit of users in a seamless manner. Adaptivity is long studied both in adaptive web systems and adaptive e-learning systems [P. Brusilovsky et.al 2007][10], and in such systems adaptivity is generally considered as an aspect between user and application based on user profiles and models.

Pervasive human-centric computing systems are proactive systems that are capable of sensing, measuring, monitoring, predicting and reacting to physical world conditions. To support a wide variety of human activities, pervasive human-centric computing systems must be:

Pervasive: being available everywhere and accessing the same information base through every portal

Embedded: sensing and affecting the physical world;

Nomadic: allowing users and computations to move around freely to meet the users' needs;

Adaptable: providing flexibility and spontaneity in response to changes in the user's requirements and operating conditions;

Powerful yet efficient: freeing itself from constraints imposed by bounded hardware resources, addressing system constraints imposed by user demands and available power or communication bandwidth;

Intentional: enabling people to name services and software objects by intent;

Eternal: never requiring shut down or reboot while components are added or removed in response to demands, errors or upgrades [MIT PROJECT OXYZEN][2].

In a pervasive computing environment, user and perceptual technologies will directly address human needs and consist of the following:

Collaboration Technologies: enabling the formation of spontaneous collaborative regions that accommodate the needs of mobile people and computations, and also provide support for recording and archiving speech and video fragments from a variety of sources and/or events;

Knowledge access technologies: offering vastly improved access to information and customized to the needs of users (i.e. people, applications and software systems);

Automation Technologies: offering natural, easy to-use, customizable and adaptive mechanisms for automating and tuning repetitive information and control tasks;

Perceptual Technologies like speech and vision technologies: enabling communication with devices, networks and software to extend the range of user technologies delivered to all places.

3. CONTEXT AWARE PERVASIVE COMPUTING

Context aware computing aims to enable device to provide better service for people through applying available context information [L.Han et.al2008][11].

Above a generic definition of context aware computing is given, which emphasizes the relation between user, context and computing, but how do we apply available context information? Although various categorizations for context-aware systems are already given [J.Pascoe1998][12], we prefer to re-interpret these categorizations based on adaptive systems, particularly according to adaptive web systems. This is because we defined adaptively as a key factor of intelligence and as a key relation between context and computing for context-aware computing systems. Therefore by referring to the field of adaptive web [P.Brusilovsky et.al2007][10] for categorization of context aware computing applications, below categorization have been proposed:

(1) *Context based filtering and recommendation of information and services*: examples might include finding the nearest printer, accessing the history of a nearby object etc.,

(2) *Context based presentation and access of information and services*: e.g. selecting voice when screen displays are not available (multimodal information presentation and user interfaces), dynamic user interfaces etc.,

(3) *Context based information and service searching*: e.g. location aware query rewriting for a search for available restaurants (query rewriting is a technique used in adaptive web systems for information filtering by rewriting a user query according to the user profiles) etc.,

(4) *Context adaptive navigation and task sequencing*: adaptive navigation is a technique employed in adaptive web systems. We can extend this idea in pervasive computing since a user's interaction might consist of several related sub-tasks in relation with his goals and might lead to context aware task sequencing,

(5) *Context based service and application modification/configuration:* this need mainly arises from different devices available in the environment, e.g. disabling particular features depending on the capabilities of target device,

(6) *Context based actions*: [J.Mantyjarvi et.al.2003][13] proposes three levels of context dependent automatic actions: manual, semi-automatic, and automatic. [11] Notes that fully automatic actions based on context are rarely useful, and incorrect action scan be frustrating,

(7) *Context based resource allocation*: this might include allocating physical recourses (e.g. memory, even non-hardware physical resources) for the use of other entities in the setting (e.g. applications, users etc.).

It is worth to note that, adaptive behaviors of context aware systems are not necessarily need to depend on the current context, rather such systems should also be able to adapt proactively by making use of current context or historical context to predict future context of the setting. An example is given in [J.Coutaz et.al.2005] where a user walks through the building and submits a printing request, the selected printer should not depend on the user's current location but rather to his final destination.

4. SOFTWARE ENGINEERING CHALLANGES

Pervasive computing has introduced new high level system requirements that should be taken in consideration in the design and implementation process.

Interoperability is highly demanded in all the levels of pervasive systems.

Software components should be built independently of the context, this way they will be used in different computing environments and applications.

Heterogeneity is a challenge in pervasive environments; mobile users interact with the system using different hardware devices, the context and connectivity become dynamic. Meanwhile, the user also has a dynamically evolving profile. As a result the software should provide services that adapt with different screen resolutions, user-interaction methods, machine power and processing capacities.

Mobility is an important requirement. Actual mobility is the capability of an autonomous software agent to dynamically transfer its execution towards the nodes where the resources it needs to access are located. Exploiting this form of mobility will save network bandwidth and increase the execution reliability and efficiency.

This aspect can be deployed to help embedded software agents to follow mobile users wherever they go. Virtual agent mobility is the ability to be aware of the multiplicity of networked execution environments [Chug ES Et.al 2004][6].

Survivability and security provide systems with powerful capacities. Survivability is the ability of a system to fulfill its mission on time despite the presence of attacks and failures. Such functioning requires a self-healing infrastructure with improved qualities such as security, reliability, availability and robustness. An application may employ security mechanisms, such as passwords and encryption, yet may still be fragile by failing when a server or network link dies. The literature has presented two kinds of survivability in the context of software security: survival by protection (SP) allows security mechanisms like access control and encryption to ensure survivability by protecting applications from malicious attacks. The survival by adaptation (SA) gives the application the ability to survive by adapting itself to the changing dynamic conditions [Chug ES Et.al2004][6].

Continuity is a very demanding feature in ubiquitous/pervasive applications especially with the uncertainty and instability of user connectivity while he moves around.

The application should have the possibility to pause the user session in the case of sudden disconnection and continue to work later without losing information[Weiser, M 1993][4].

Usability, according to human-machine interaction engineers and to interactive system designers, is how easy an interface design is to be understood and used, how unambiguous interactive controls are and how clear its navigational scheme is [Gschwind T 2002][9]. This feature is the final objective of pervasive systems.

5. CONCLUSION

To make pervasive computing a reality, a significant amount of research is still needed; much of it in software engineering. To ensure that pervasive computing systems fulfill their required and intended purpose and can trust these systems to perform as intended Software is an essential ingredient. To produce pervasive computing systems efficiently and economically we need new methods and techniques to meet the new challenges. Our techniques must be able to deal with highly dynamic, highly heterogeneous environments and scale the range of tiny to powerful computing elements.

6. REFERENCES

[1] Dertouzos, M.L., *The Unfinished Revolution: Human-Centered Computers and What TheyCan Do for Us*. New York: HarperCollins (2001).

[2]. MIT Project Oxygen, http://www.oxygen.lcs.mit.edu/

[3]. Weiser, M., The computer for the twenty-first Century. *Scientific American*, 265, **3**, 94-104, 1991

[4]. Weiser, M., Some computer science issues in ubiquitous computing. *Comm. ACM*, 36, **7**, 75-84,(1993).

[5] Chen E, Zhang D, Shi Y and Xu G, "Seamless Mobile Service for Pervasive Multimedia". In PCM'04, IEEE, 2004, p 194-198.

[6] Chung ES, Hong JI, Lin J, Prabaker MK, Landay, JA and Liu AL, "Development and evaluation of emerging design patterns for ubiquitous computing". In 2004 conference on Designing interactive systems, USA, p 233 – 242,2004.

[7] Davis J, Tierney A and Chang E, "A User-Adaptable User Interface Model to Support Ubiquitous User Access to EIS Style Applications". COMPSAC'05, IEEE, p 351–358,2005

[8] Duan Y and Canny J, "Protecting User Data in Ubiquitous Computing: Towards trustworthy environments". In PET 2004, Springer, p 167-185,2004.

[9] Gschwind T, Jazayeri M and Oberleitner J, "Pervasive Challenges for Software Components". In RISSE 2002, Springer, p 152-166,2002.

[10] P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *The Adaptive Web*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 2007.

[11] L. Han, S. Jyri, J. Ma, K. Yu, “Research on Context-aware Mobile Computing”, *Proceedings of Advanced Information Networking Applications Workshops*, pp 24-30, 2008.

[12] J. Pascoe, “Adding generic contextual capabilities to wearable computers”, *Proceedings of 2nd International Symposium on Wearable Computers*, pp. 92-99, 1998.

[13] J. Mäntyjärvi, U. Tuomela, I. Kansala, J. Hakkila, “Context-studio–tool for personalizing context-aware applications in mobile terminals”, *Proceedings of Australasian Computer Human Interaction Conference*, Addison-Wesley Longman, pp. 64-73, 2003.

[14] P. Korpipää, J. Hakkila, J. Kela, S. Ronkainen, I. Kansala, “Utilizing context ontology in mobile device application personalization”, *Proceedings of Mobile and Ubiquitous Multimedia*, ACM Press, pp. 133-140, 2004.

[15] J. Coutaz, J. Crowley, S. Dobson, D. Garlan, "Context is Key", *Communications of the ACM*, 48(3), March 2005.