



Design and development of context- Aware and context- Adaptable system with session handling mechanism for ubiquitous computing

S. Sahana^{1*}, A.B. Karthick Anand Babu² and G. Kumaravelan³

Abstract

An extension based on Ubiquitous computing in smart environment and future of this smart environment is to be enhanced by two mechanisms namely Context-awareness and Context-adaptability. These mechanisms will play a significant role to offer valuable services to the end users, it will also provides ample of challenges and opportunities for the application developers to increase the system behavior and usability in ubiquitous computing. In order to integrate the context-awareness and context-adaptability in the existing systems with an objective of increasing the performance as well as to support the application developers to design many applications for smart environment, this paper proposes architecture called "CAAS Framework".

This proposed architecture consists of Logic unit, control unit, mapping unit, protocol unit, object unit and monitoring unit. This proposed architecture is designed as light-weighted plug-in to support existing frameworks. Two case studies are performed with this architecture; one case study is to test with all the units as a single plug-in in learning management system and the other case study to test a specific unit (monitoring unit) in CAAS framework. From the case studies, we observed that the proposed architecture work superior compared with an existing system in terms of usability and its performance.

Keywords

Human Computer Interaction (HCI), Ubiquitous Computing (UC), Context-aware, Context-adaptable, Session handling.

AMS Subject Classification

34L30, 15A06, 62M10, 62J02.

^{1*}Research Scholar, Department of Computer Science, Tamil University, Thanjavur-613010, Tamil Nadu, India.

²Department of Computer Science, Tamil University, Thanjavur-613010, Tamil Nadu, India.

³Department of Computer Science, Pondicherry University, Puducherry-605014, Union Territory of Puducherry, India.

*Corresponding author: 125sahana@gmail.com

Article History: Received 11 November 2020; Accepted 29 December 2020

©2021 MJM.

Contents

1	Introduction	84	3.6	Logic Unit	86
2	Related Work	84	3.7	General Interaction Protocol (GIP)	86
3	This Framework Elaborate Existing Dandelion Framework and its Drawbacks	85	3.8	Final Interaction Objects (FIO)	86
3.1	Proposed Methodology	85	3.9	Kinetic Sensor	86
3.2	System developer	85	3.10	Session handling Mechanism	86
3.3	System Installer	85		Actions performed by the Session Service:	
3.4	System Unit	85	3.11	Session Timeout	86
3.5	UI control Unit	86	4	CASE STUDY: Learning Management System in College	87
			5	Conclusion and Future Work	87

References 87

1. Introduction

In the era of the 21st century, the development of technological creation is enormous in the multidisciplinary field of Human-computer interaction (HCI) as well. HCI is an extremely wide control that encompasses various specialties with various concerns regarding computer development. It is an innovative branch of computer science that deals with the way machines and individuals interface.

Human computer interaction is the advancement about an enhancement of technology and the efficacy of the devices. Nowadays the interaction is beyond time, devices and location leading to the concept of “ubiquitous computing”.

Ubiquitous computing (UC) afford the framework to achieve real information and its extensive processing in an infinite way. Ubiquitous computing is the assemblage of controlling, computing and communication technologies. UC perform the task disregarding of the time, the situation, and all alternative variables in the context. Ubiquitous computing aims to frame an illstrate application placed on the surrounding of user context to support frequent activities.

In the ubiquitous application the context would perform a core part, which enables to manage the implied interaction even though there are still unsolved problems relating to the acquisition of context, portrayal of context, and its distribution.

The System of context-awareness is aware of its surroundings, context or situation so that the mechanism can adjust to the situation and deliver services based on the modified situation or context[1]. Consequently, the authorization decision-making process should be capable of capturing various and dynamic contextual and situational information and incorporate it into devices, users and resources for dynamic access management. Context adaptability refers to users that can substantially customize the system through tailoring activities by themselves, i.e. an adaptable system.

Over the past few years, a variety of application scenarios have been presented where contextual knowledge, such as smart phones, plays a major role[2]. In this type of environment, an important task is the recognition process of a resident’s activity in the environment to anticipate the needs of occupants and to adapt to such a intelligent environment. In any artificial environment, there can be differences between the intent of its architecture and the manner in which it is used.

In this paper we address the common issues across to an existing framework of UI Dandelion mechanism for the growth of Ubiquitous Computing that successfully upgrade the security of those systems between different environments[3].

This is accomplished by including the new method of identifying sessions into an existing architecture for a specific combination of user and environmental features. Using fixed time limits for session identification, we use a session timeout approach which produces stronger results to identify session limits.

The rest of the paper is categories as accompany of below: In Section 2 provides a review of associated work in the discipline. Section 3 we give an initiation to the CAAS framework together with a preliminary examples of the implementation of specific features and attributes are using it. The Comparative Study of analysis is carried out in the Section 4, both from an existing and proposed framework. In the end of the paper, Section 5, we present some of the conclusions and future work.

2. Related Work

It is an efficiently acknowledged that one of the barrier in the way towards accomplishing accurately ubiquitous systems able of operating in a multiplicity of scenarios, stretch carry out a tolerable measure of user occurrences, is that it is entirely burdensome to foresee, and produce support for, the broad span of situations, settings, device equipment and technologies (resources of user interaction) vital to construct User Interaction’s able of as long as a requisite of session handling mechanism in a variety of use of a contexts.

In this work, we focus the issues and the specific realization of the user interaction for a particular outline (characteristics of user and environments, devices, modalities, etc.). The objective is to upgrade the convenience of the system of ubiquitous computing, as the result that, they can be effortlessly adjustable to variety of contexts.

In[4], Dandelion, a unified code offloading system for wearable computing, is introduced. The author suggested work to take advantage of nearby computing resources by D2D and cloud offloading. The author model can significantly increase computing power and reduce the energy consumption of portable devices for Android OS such as Google Glass. In future buddy researchers can investigate the feasibility to migrating the system to other platforms as well as security and confidentiality issues aimed at integrating them into the next generation design.

The authors[5] seek to develop a new open platform for activating context-sensitive Service Front-Ends (SFEs) for the UI. Authors establish the SERENOA framework that increase user satisfaction and performance relative to traditional SFEs based on manually designed UIs. Authors model show the car rental abstract ASFE-DL SERENOA language and then versions adapted for different devices can be derived from it. SERENOA enters now in the critical stage for its standardiza-



tion objectives: a W3C Working Group on Model-Based UIs was established has started collaborating with the consortium.

Author[6] focused the concept of abstraction technology for user interaction devices in the context of Ambient Intelligence and Ubiquitous Computing. Authors developed a dandelion framework with multi-agent technologies to build a physical UI's based on the GIP. Authors carried out the framework techniques of model-driven approach with distributed device abstraction to develop of physical UIs. Authors model outcomes are to reduced the coupling between the systems and end devices for implement the physical UIs. From this study, author suggest to the upcoming researcher to use the system and device independently.

Authors[7] deal with the domain of Ubiquitous Computing Systems and Ambient Intelligence Provide support for user interface development. The authors outline the Dandelion framework for a model-driven approach to defining a range of top-level reporting models. The results of the author's model dissociate the logic of the application, the state of the environment and the state of the user's interacting elements. The author suggests to the buddy researcher to change and dynamically adapt its user interface easier.

3. This Framework Elaborate Existing Dandelion Framework and its Drawbacks

Dandelion is a conceptual structure intended to weaken the unpredictability and expenses of creating programs which backing the natural interaction with the ubiquitous activities. The objective of the Dandelion structure lies in two viewpoints that are to give a transparent and basic coherent system for program designers and also to contribute an efficient and robust framework for the clients. Its principal goal is to afford a reference implementation of the proposals and conceptual framework with the point of demonstrating the proposed work of the "CAAS framework" Context-Aware and Adaptable System.

In the existing system of UI [3], the dandelion framework has met several issues and constraints, from that we addressed the one main issue of the design. In the existing framework the mechanism of session handling process is not carried out to get the user information and tracks the details of the logged page of the user component for the particular web pages.

3.1 Proposed Methodology

Right now, we propose another structure to address these issues by expanding the level of the user's context and to optimize the resources effectively in the middle of the code of User Interaction, the device, and the user resources. The general proposal is initiate a latest structure for Ubiquitous Computing UI evolution, which tracks out the logged details

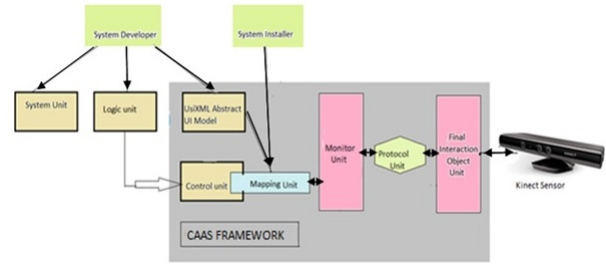


Figure 1 Represent the Context Aware and Adaptable System Framework

and the session data of the user access of the different web pages. In spite of this, we introduce only the main concept of session timeout mechanism only from the session handling approach this framework is called "Context Aware and Adaptable System (CAAS) Framework".

Figure 1 represents the software framework of CAAS Context Aware and Adaptable System. Within the CAAS framework was designed as a benchmark implementation of the design architecture and user interface development architecture for Ubiquitous systems, promote the design layout and implementation of the session management mechanism.

In this CAAS framework that contain the system developer, system installer, they are consist of the System unit, logic unit, UsiXML Abstract UI model, control unit, mapping unit, monitoring unit, protocol unit, final interaction object unit and finally the gesture recognition kinect sensor.

3.2 System developer

CAAS permit to system developers can describe and sketch the User Interaction's at the abstract level of utilize the User Interface eXtensible Markup Language (UsiXML) , and then implement the utilization of User Interface control unit on peak of the Abstract User Interface model, then it mapped into the mapping unit that have the interaction of other units in the framework.

3.3 System Installer

The system installer only know the concrete interaction resources used for effectively decoupled the UI designer and the UI control unit. System developers/ installer only specify the interaction devices which will be implemented. The system installer only has the rights to mention the various mapping technique for each phases.

3.4 System Unit

The logic of the system and the interacting resources function in a distributed fashion, with CAAS in the foundation of their decoupled. It controls one or a number of interacting



resources to produce certain data to the user. The logic of the system needs interactive resources to attract a focus of the selection and attention to the user.

3.5 UI control Unit

The CAAS enables programmers to conceive UI at the declarative and abstract level adapting the UsiXML Abstract UI template, and then implement the UI control logic in addition to these abstract perceptions. As a consequently, the user interface designer and user interface control logic are actually decoupled from the real-world interaction resource utilized.

3.6 Logic Unit

The logical unit is effectively decoupled from the actual interaction resources used, which only the system installer should be aware of. The developers conceive and construct the logical unit over the abstract concept rather than specific components of the actual technology and the specific API of the interactive resources used for every scenario.

3.7 General Interaction Protocol (GIP)

The General Interaction Protocol (GIP) provides a shared interface for remote-accessible interaction operations that can be accessed remotely via a network protocol. It is a distributed interaction protocol intended to imitate the concept interaction operations underpinned by the abstract interaction pattern. This convert the co-operation activity to achieve by the logic unit on peak of generic interaction element into events in the GIP protocol that are sent over the network to the remote equipments that implement the real user interaction, this component is known as Final Interaction Object (FIO). It distributed device abstraction layer that encapsulates the specific behavior of each device behind a generic interface of user interaction operations. This GIP implementation is based on publication/subscription messaging technologies, where all the interaction actions supported by the GIP is defined as published tasks that are received by distributed Interaction Resources.

3.8 Final Interaction Objects (FIO)

The FIOs are the elements responsible in charge of ultimately connecting CAAS with the actual devices and software components which carry out the interaction with the user. To support the wide variety of IRs available, the CAAS is required to rely on a many values of different FIO implementations. UniDA makes it possible to develop generic FIOs able to use devices from different manufacturers, with different APIs and protocols.

3.9 Kinetic Sensor

Kinect is a motion sensing input devices for free natural user interaction with a system or devices. This device contains the following menus that are, Multi-array microphone, Color VGA video camera and Depth sensor.

3.10 Session handling Mechanism

In the mechanism of Single Sign-On (SSO) infrastructure include the centralized component of Session Services. The session management mechanism enables the retention of authenticated user session information in several applications involved in the same SSO environment. The Session service also to chase out the interaction of users with web-based applications, to furnish basic administrative and surveillance capabilities for user session managements.

3.10.1 Actions performed by the Session Service:

- Identifier of session is generate
- Session state information is maintained
- Implemented the session behavior of time-dependent variable
- Session life cycle events implementation (e.g. creation of session, destruction of session, etc.)
- Provoke event of session state change notifications / session life cycle so that all members in the equivalent Single Sign-On (SSO) status can get notified.
- Web application permits Single Sign-On (SSO)
- Grant to share the data by participating clients across the formations
- Provide Interface for managing the session
- Implement the facilities
- Tracking the Session

In this paper we mainly were focusing the concept of “Session timeout” mechanism only from the broad area of session handling.

3.11 Session Timeout

The centralized area of the Session handling concept is Session timeout mechanism and it also has a component of security enrichment of the performance. Whenever the client interacts with the web page, this timeout countdown will reset. To protect an idle clients are logged out immediately for significantly diminish device exposure to data breach. Force-response timeout, triggered after a convinced duration of the time.



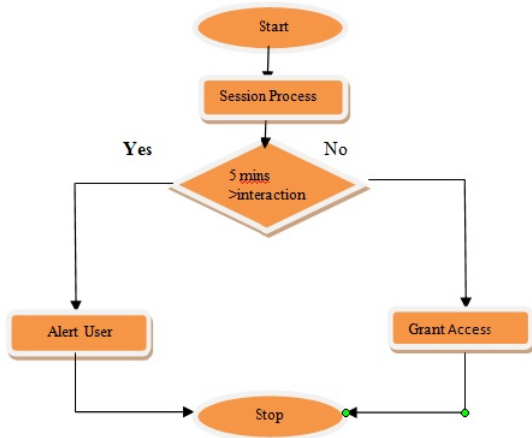


Figure 2

Represent the flow diagram of a MOODLE session handling.

Regarding of client a session mechanism is conserve a service information of how long-lasting has been a device is logged in, and carries out the limits of timeout, when precondition.

- After an inactive of user for a certain duration of time , your site automatically sign out, that are made by the session timeout to match that period of moment.
- Prolong the time of session, if you have a much amount of content and user experts to take a part for a long duration engaging with that content.
- Contrarily, minimize the session time out, if the site has a limited number of content.

4. CASE STUDY: Learning Management System in College

In this section, the case study constructed on learning management system in college is discussed in detail. The familiar Learning Management System of free and open source Modular Object Oriented Environment (MOODLE) is used for this study.

Figure 2 is represents the flow diagram of a MOODLE session handling mechanism. Initially start the process, then in the next step of flow move to the session process. Meanwhile the session process is beginning, when the users sign in to their MOODLE account. The session handling mechanism allows the server to automatically sign a user out after being inactive for a certain amount of time, here we give the minimum time is 5 minutes. With that process it track students logged information and as they access the data of different web pages by grant the permission of further use. In case that the users do not load a new page in the specified period of time, MOODLE will end their session and disconnect them with the website

administrator through the alert information. The overall rights of handling the mechanism such as setting the time of session, tracking the session details are handled by the MOODLE administrator.//

5. Conclusion and Future Work

We proposed in this paper that the “CAAS framework” improves the performance of session handling concept especially in session timeout mechanism and records the log details in efficient way. To increase the level of user interaction device by using the kinect sensor for gesture reorganization, it provides the exact result outcomes of the work. In the future, we envision the research activity is to look deeper into the broad topic of the session handling other mechanism of tracking the details, file handling and process analysis in various domains of applications.

References

- [1] Edwin JY Wei and Alvin TS Chan. Towards context-awareness in ubiquitous computing. In *International Conference on Embedded and Ubiquitous Computing*, pages 706–717. Springer, 2007.
- [2] Nuhcan Akçit, Emrah Tomur, and Mahmut Onur Karslıoğlu. Context-awareness in ubiquitous computing and the mobile devices. In *Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015)*, volume 9535, page 95350R. International Society for Optics and Photonics, 2015.
- [3] Gervasio Varela, Alejandro Paz-Lopez, Jose A Becerra, and Richard Duro. A framework for the development of context-adaptable user interfaces for ubiquitous computing systems. *Sensors*, 16(7):1049, 2016.
- [4] Morteza Golkarifard, Ji Yang, Zhanpeng Huang, Ali Movaghar, and Pan Hui. Dandelion: A unified code of-flooding system for wearable computing. *IEEE Transactions on Mobile Computing*, 18(3):546–559, 2018.
- [5] Javier Caminero, María del Carmen Rodríguez Gancedo, Jean Vanderdonckt, Fabio Paternò, Joerg Rett, Dave Raggett, Jean-Loup Comelieu, and Ignacio Marín. The serenoa project: Multidimensional context-aware adaptation of service front-ends. In *LREC*, pages 2977–2984, 2012.
- [6] Gervasio Varela, Alejandro Paz-Lopez, Jose A Becerra, and Richard J Duro. The generic interaction protocol: Increasing portability of distributed physical user interfaces. *Romanian Journal of Human-Computer Interaction*, 6(3), 2013.
- [7] Gervasio Varela, Alejandro Paz-Lopez, Jose Antonio Becerra, and Richard J Duro. Dandelion: decoupled distributed user interfaces in the hi 3 ambient intelligence platform. In *International Conference on Ubiquitous Com-*



puting and Ambient Intelligence, pages 161–164. Springer, 2012.

ISSN(P):2319 – 3786
Malaya Journal of Matematik
ISSN(O):2321 – 5666

