A Design of CBD Intelligent Framework for U-Healthcare Application

Yvette E. Gelogo\textsuperscript{1}, Haeng-Kon Kim\textsuperscript{1} and Youn-Ky Chung\textsuperscript{2*}

\textsuperscript{1}School of Information Technology, Catholic University of Daegu, Korea
\textsuperscript{2}Department of Computer Engineering, Kyungil University, Korea
\textsuperscript{1}yvette@cu.ac.kr, \textsuperscript{1}hangkon@cu.ac.kr, \textsuperscript{2*}ykchung@kiu.ac.kr

Abstract

Distributed software agents offer great promise in building an increasingly pervasive middleware and component technology. Context-aware systems are able to monitor and automatically adapt their operation accordingly to the execution context in which they are introduced. Component-based development (CBD) focuses on the development and reuse of self-contained software assets in order to achieve better productivity and quality. This study focuses on the design of the framework which supports an active repository in developing a component. To test the proposed CBD intelligent framework, the u-healthcare monitoring system has been presented.

Keywords: CBD, context-aware systems, u-healthcare, active repository

1. Introduction

Context-aware systems are able to monitor and automatically adapt their operation accordingly to the execution context in which they are introduced. Component-based development (CBD) focuses on the development and reuse of self-contained software assets in order to achieve better productivity and quality. In order to store and retrieved components, CBD employs component repository systems to provide components to the system developers. This paper presents an active component repository that is able to receive the current configuration from the context-aware system and compute the components and the new architecture that better fit the given context. Since the repository has a wide knowledge of available components, it can better decide which configuration is more suitable to the running system [1, 4].

Ubiquitous healthcare is an emerging technology that promises increases in efficiency, accuracy and availability of medical treatment. The purpose of u-healthcare monitoring system is to provide convenient healthcare service to both caregivers and patients, and to make it easy to diagnose patient's health condition. People can monitor their health without visiting the hospital or clinic [6].

Distributed software agents offer great promise in building an increasingly pervasive middleware and component technology. Agents are specialized kinds of components that offer great flexibility than traditional components. This study focuses on the design of the framework which supports an active repository in developing a component. To test the proposed CBD intelligent framework, the u-healthcare monitoring system has been presented [1].

This paper is organized as follows: Section 2 discuss the background of the study which includes the terms and technology related to the study. Section 3 present and discuss the proposed CBD intelligent framework. Section 4; discuss the u-healthcare case
study implementation. Lastly, the conclusion and future works.

2. Background of the Study

2.1 Component Based Development

One of the tasks of the software architect is to produce the list of components that will populate the architecture. This list gives the development, mining, and acquisition teams their marching orders for supplying the parts that the software system will comprise.

By component development, we mean the production of components that implement specific functionality within the context of a software architecture. The functionality is encapsulated and packaged and then integrated with other components using an interconnection method [1, 5].

2.2 Context-aware

Context awareness is a property of mobile devices that is defined complementarily to location awareness. Whereas location may determine how certain processes in a device operate, context may be applied more flexibly with mobile users, especially with users of smart phones. Context awareness originated as a term from ubiquitous computing or as so-called pervasive computing which sought to deal with linking changes in the environment with computer systems, which are otherwise static. The term has also been applied to business theory in relation to Contextual application design and business process management issues [4].

In computer science context awareness refers to the idea that computers can both sense, and react based on their environment. Devices may have information about the circumstances under which they are able to operate and based on rules, or an intelligent stimulus, react accordingly.

2.2.1 Context-aware Application in u-healthcare: Context-aware mobile agents are a best suited host implementing any context-aware applications. Modern integrated voice and data communications equips the hospital staff with smart phones to communicate vocally with each other, but preferably to look up the next task to be executed and to capture the next report to be noted.

However, all attempts to support staff with such approaches are hampered till failure of acceptance with the need to look up upon a new event for patient identities, order lists and work schedules. Hence a well suited solution has to get rid of such manual interaction with a tiny screen and therefore serves the user with

- automated identifying actual patient and local environment upon approach,
- automated recording the events with coming to and leaving off the actual patient,
- automated presentation of the orders or service due on the current location and with
- Supported documenting the required information keying in a minimum of data into prepared form entries.

2.3 Active Repository

The active repository is capable of providing information to the users by monitoring their development activities without the need to receive explicit queries from them. These systems execute in background inside an integrated development environment (IDE) monitoring the user activity and suggesting possible software components to be used in the current development context. Both the traditional and the active repositories provide functionalities only during the development phase of the software system. Once the system is deployed, the repositories are of no use [8].

This way, in order to promote the runtime adaptation of context-aware component-based systems, this paper presents an active repository that actively provides new
software components that are more adapted to the context of the adaptable system. Since the repository has a wide knowledge of the available components, it can better decide which configurations of components are more suitable to the running system.

In the proposed approach, the context aware system informs to the repository its current configuration and its context information and the repository is able to compute the components and new architecture that better fit the given context. In this way, the component repository commonly adopted in component-based software systems is expanded to provide components, not only during the development stage but also context aware components during the operational stage of the system life cycle.

3. Proposed A CBD Intelligent Framework

3.1. System Architecture

To allow the construction of CBD Intelligent Framework, we have created a framework named CBD-IF (CBD-Intelligent Framework). This framework is divided into two components, the client and server components that will handle the client system context changes and generate new system architectures over the time through component reconfiguration. Figure 1 shows CBD-IF’s general architecture. This framework is not intended to be restricted to a specific component model. However, to use CBD-IF’s services, the client system needs to be developed using a component model that allows at least the simple reconfiguration commands (add, remove, replace, connect, or disconnect a component in the system). In addition, the client system must implement some modules that allow CBD-IF retrieve the system current configuration state and context representation [1].

CBD-IF’s modules can be divided into two disjoint groups, the Client components and the Server components. As shown in Figure 1, the Client components are as follows.

A. **Monitoring manager**: The monitoring manager is responsible for monitoring the environment, if there is a presence of new context, the change of user’s location, the available components based on the received context.

B. **Execution Manager**: responsible to combine information collected by the Monitoring Manager in a meta-representation that will be used to generate a new optimized configuration to the system.
C. **Configuration Manager:** This is responsible for reading the new context-aware information received from the monitoring manager and instantiate new architectures sent by the CBD-IF server, new architectures will be instantiated by adding, removing, or replacing components or interconnecting existing ones.

D. **Repository Interface:** provides communication with CBD-IF server. This module is able to provide the client system an abstraction of the server location in order to enable the use of the distributed services.

The CBD-IF server provides a set of services for receive client system context and provide reconfiguration commands if needed. The server components are listed below.

A. **Evaluator:** Evaluate the context-aware information and search for appropriate component that is stored in active repository.

B. **Component Storage:** stores components into the repository.

C. **Component locator:** Responsible for locating the components in the distributed repository.

D. **Optimizer:** Implements heuristic algorithms to solve the best configuration problem.

E. **Client Interface:** Implements the communication between the clients and the server.

3.2 **Overall Architecture**

3.2.1 **Monitoring Manager Sub-module:** The process starts with context data received through physical sensors, virtual sensors, and explicit user actions.

Almost all mobile devices are equipped with some form of wireless network technologies such as GSM, GPRS, UMTS, Bluetooth, Wi-Fi, Radio Frequency, IrDA, etc., and can therefore sense if there is a network connection around them. Moreover, the device might be equipped with sensors capable of sensing data about the physical world surrounding the mobile device for example noise, light level, temperature, etc. The device might be able to receive data about its environment such as temperature, etc. from some surrounding sensors. “Virtual” sensors receive data from other processes running on user’s mobile device, like an agenda, a timer, an alarm clock, and so on. For explicit user actions, the user can explicitly communicate, via the user interface, data about the current context. For instance, he/she might choose a connection/ network provider; set the alarm clock, select the silent mode, and so on [1, 5].

![Figure 2. Client Component](image)
3.2.2 Configuration Manager: The Configuration Manager submodule is in charge of selecting which context to download and to download their code. Its activity is triggered by notifications of context entry and exit events, received from the Context submodule. The Filter and Download Scheduler receives these notifications and, on the basis of its internal criteria, also depending on user’s preferences, decides when to request the current public context descriptors to the Context submodule and to forward them to a Descriptors Server (DS). The DS is in charge of selecting, on the basis of the received context descriptors.

Since not all the context selected on the basis of the public context descriptors will be downloaded (nor executed), the MDS does not store and send context code, but just context descriptors. Each descriptor is a simple XML file that contains several structured and unstructured data about the corresponding context: a unique identifier, a textual description, a manifest declaring which resources the context will need and use while executing, a download server from which the actual context can be downloaded, and so on. [1, 5].

3.2.3 Executor Sub-module: The last sub-module of the pipeline is the Executor. Its aim is to run each downloaded CBD-IF inside a Sandbox, in order to avoid malicious context to use resources against user’s will. Each context is managed by the Scheduler, which is capable of starting, pausing, stopping, and destroying the context-information. The Scheduler is notified of context exit (and entry) events, to stop those context that go out of context. Each context can register itself with the Registry inside the Context submodule, in order to be directly notified of relevant context change events.

Each context that has to use resources outside its sandbox is allowed to do so only through the Security Manager, which denies requests that are incompatible with context manifest and prompts the user to confirm more heavy resource usages.

3.3 Software Support

Servlets are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms, servlets are server- and platform-independent.
4. U-Healthcare Case Study

To test the designed framework, this case study for u-healthcare application was conceptualized.

Ubiquitous healthcare is an emerging technology that promises increases in efficiency, accuracy and availability of medical treatment. The purpose of u-healthcare monitoring system is to provide convenient healthcare service to both caregivers and patients, and to make it easy to diagnose patient's health condition. People can monitor their health without visiting the hospital or clinic.

Ubiquitous healthcare systems will monitor patients as they maintain their normal everyday activities, in order to warn the patients or healthcare workers of problems as well as collecting data for trend analysis and medical research. The continuous monitoring of the health record would give a better diagnosis. The integration of wireless body sensor network is very important for timely monitoring.

In this case study, the information from the sensors, like heart rate, vital signs, and location of the patient is being monitored. The context information is gathered using mobile devices and this context information will be processed.

To slowly explain this case study, first we discuss the u-healthcare system and its components. In the later part, I will discuss the integration of my proposed framework.

4.1 U-healthcare Background

4.1.1 U-healthcare System: Smart, easy to use, cloud-connected and multi-function medical health sensors that can capture a rich set of physiological and biological parameters are now a reality and bring medical monitoring to a whole new level. These advances will revolutionize the medical practice of preventive care and foster a new era in medical devices. Algorithmic, circuit, and form factor innovations, coupled with integration and miniaturization of sensors, along with intelligent usage feedback mechanisms using devices are the driving forces behind the technology advances in medical monitoring. Quality of care can now be drastically improved by use of these sophisticated at point-of-care devices, serving a multitude of medical market segments such as (i) elderly managed care, (ii) chronic disease care, (iii) hospital discharge care, (iv) telemedicine, to even (v) average consumers who want to be empowered to better understand their health in collaboration with their physicians and care providers [6].

4.1.2 U-healthcare and Mobile Technologies: Many healthcare providers are adopting mobile devices successfully in a diverse range of practices. Several reasons make mobile devices attractive to healthcare providers. The primary reason is that many POC mobile device apps have been developed. The second reason is that healthcare providers are mobile workers and a mobile device provides a huge amount of information at their fingertips no matter where they are. Even when a healthcare provider is in his office, he is moving around a lot, not really stationed at his computer. Further, the multiple media aspects of mobile devices, along with connectivity and mobility can help a healthcare provider become more efficient. Meanwhile, patients are also getting used to accessing health information, actively participating in their own healthcare, and maintaining contact with their healthcare providers through mobile devices. Therefore, mobile devices are starting a new epoch in healthcare, and there are significant opportunities to exploit the potential of mobile devices in healthcare.

4.2 Integration of CBD Intelligent Framework for U-healthcare Monitoring System

The context is generated from the patient environment and sends to the mobile device. The mobile device process the receive data as context. These contexts information will then be process and send to the server. The server read the data and search for the
appropriate component from the repository. For example, the sensor send the signal of vital signs of the patient and detected to be in trouble, then the context aware information will be send from user’s mobile device to the nearest hospital or call the emergency number to rescue the patient. This context-aware information on the process is raw data which needs to be process and transform into information that the person can understand. Therefore to efficiently process these data, the designed framework which is an active repository for running system will be used. By using the framework, the information gathered from the environment and the patient condition with the help of sensor devices, will be process using the designed component based development and send to the user or the hospital as organized information. This information can be in form of the following:

a. Alert Message Services
   a. Low Danger
   b. Medium Danger
   c. High Danger

b. Location Information Services

c. Condition of the Patient Services
   The repository applies Fuzzy logic algorithm to evaluate the adequacy level of the components and GRASP algorithm to mount the new system architecture.

5. Case Study Implementation

This section discuss the Mobile, Network Technologies and software support to implement the u-healthcare system monitoring case study.

5.1 Mobile and Network Technologies

Continuous health monitoring is essential for recuperating patients and patients with chronic health conditions. It can be used in judging whether an individual is in a physiological state suitable for his duties, and it is also useful for people of all age groups who have possible health problems or have opted for a healthy lifestyle.

![Figure 4. Mobile and Network Components](image)

The sensor devices gather the context-aware information. The communication protocol between PHDs (Personal Healthcare Device), CE (Computing Engine) and MS (Monitoring Server) are very important. With this, we designed a multi-purpose gateway using ISO/IEEE 11073 between PHDs and CE modeled by design preferences. The PHDs send message to CE one at a time is real application. However, CE should be capable of receiving multiple messages from PHDs. Therefore, we design a multi-purpose gateway that can handle multiple message transmission from PHDs. The challenge is to work on the efficiency of the transmission and design preferences are applied in order to map it dynamically. Figure 5 shows the General Overview of the u-healthcare system. Figure 4
shows the mobile and network components [9].

In this paper we also consider the communication between the CE and MS. IEEE 805.15.6 is the applied transmission protocol between CE and MS for WBAN network.

![U-Healthcare Monitoring Systems](image)

**Figure 5. U-Healthcare Monitoring Systems**

The mobile gateway should be hosted on in small and portable device such us smartphone or PDAs. The server side HTTP connects via GPRS, 3G, or 4G enabled mobile internet access. The first and second layer performs the translation between HTTP and Bluetooth communication which run in smartphone in J2ME application and between Bluetooth and Zigbee communication.

The data captured and measured from the wearable monitoring device is further transmitted to the Android smartphone. The Smartphone has developed apps to monitor the data transmitted from the wearable monitoring device. Bluetooth is the connection used to receive the data from the wearable monitoring device. The raw data from the wearable monitoring device and context information are encoded to Android apps. This will be process and real time monitored in mobile device. There are two main functionality of this system. First, the transmitted raw data from wearable monitoring device is send to the clinical specialist that may require a precise description of the recorded information. This can also be needed in the future diagnosis of the patient health status. Second, the presentation of the health status to the users, there are health status indicator for each vital data (Heart rate, Body temperature). From these data, the user may react for example by following the clinical guidelines or recommendations. The app is automatically connected to the healthcare provider. There is another feature that if the health status is determined to be fatal, the app will automatically notify the patient and the hospital for emergency. This information could be (1) alert message, (b) location information, and (c) condition of the patient. Figure 5 depict the u-health monitoring system [6].

### 6. Conclusion and Future Works

This paper presents an active component repository that is able to receive the current configuration from the context-aware system and compute the components and the new architecture that better fit the given context. Since the repository has a wide knowledge of available components, it can better decide which configuration is more suitable to the running system. In order to test the validity of the proposed framework, the u-healthcare monitoring application case study is adopted. Several algorithms are presented to test the efficiency of the proposed framework and the case study. In our future works, we will present the detailed implementation of the framework to our case study.
Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (IITP-2015-H8601-15-1007) supervised by the IITP (Institute for Information & communication Technology Promotion). This research was also supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant number: K 2013079410).

References
