

Service Provisioning to Mobile Devices using Context-Awareness

Chitra G. Sahu^{#1}, Dr. D. S. Adane^{#2}

^{#1}Research Scholar,

^{#1}Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India.

^{#2}Head of Department,

^{#2}Department of Information Technology, Shri Ramdeobaba College of Engineering and Management, Nagpur, India.

ABSTRACT— In today’s fast growing world, where we could find developments in every sections to enhance functionality of mobile computing devices. We hardly find any features in mobile computing devices, to make the device smarter internally by adapting itself according to the current context. Driven by the proliferation of mobile and pervasive computing, there is a great demand for context-aware, self-adaptive services provided by the middleware. Such middleware benefit users by dynamically adjusting their offered services to the highly dynamic context which characterizes mobile computing environment. To achieve this kind of sophistication, however, such middleware must be capable of sensing the context and autonomously reacting upon their knowledge on it.

In this paper, we have proposed the new approach of genetic algorithm to be used with the concept of constraint satisfaction to achieve dynamic reconfiguration of context-aware middleware. We have analyzed the current requirements of mobile computing systems and tried to achieve few of them by providing services that can better utilize the scarce resources of mobile devices. In this paper we are presenting few services offered by our middleware to resolve the limitations caused due to availability of scare resources. These services enhance the functionality of mobile computing devices.

Keywords— Mobile Computing, Context-Aware Middleware, Genetic Algorithm.

I. INTRODUCTION

Mobile computing is a generic term describing ability to use the technology to wirelessly connect and use centrally located information and/or application software through wireless computing and communication devices. The availability of lightweight, convenient computers and wireless technologies has created a new class of applications called mobile applications, presenting challenging problems to the designers. These applications often run on platforms such as PDA, notebooks, and mobile phones, each of which has limited processor power, memory, and battery life. It is either impossible or too costly to supplement the resource availability. Thus the middleware should be designed to

achieve optimal resource utilization. To assistance designers building mobile applications the research work in the field of middleware systems has increased. Middleware aims at smoothing communication and coordination of distributed components, covering complexity raised by mobility from application developers as much as possible.

However, under the highly variable computing environmental conditions that characterize mobile platforms, it is believed that prevailing traditional middleware systems are not capable of providing adequate support for the mobile computing environment. The research work by Adane D. S., Sathe S. R. and Moghe C. S., [1] has identified four categories of middleware. Each of these category aims to support at least one of the requirements imposed by mobility using different approaches. The following table illustrates various requirements met by the different categories:

TABLE I
STRENGTH AND WEAKNESSES OF DIFFERENT MIDDLEWARE TECHNOLOGIES

Requirements	Reflective	Tuple Space	Context-Aware	Event-Based
Synchronous/ connection based	X		X	
Asynchronous/ connectionless		X		X
Re-configuration	X			
Adaptation	X		X	
Awareness	X		X	
Light weight				X

Based on their work we found that there is scope for providing dynamic reconfiguration to the context aware middleware. Also with the advent of context-aware middleware many issues came out, like security [5], balance of user control [6] and threat to privacy of the user [4]. Thus

there is a great demand for designing modern context aware middleware systems that can support new requirements imposed by mobility. As context aware middleware deals with the current context, it is the only middleware that can meet the changing requirements imposed by mobility as it can adapt itself according to the changing context. We can overcome these issues with the new middleware approach.

In this paper Section II discusses the related work. Section III covers the partial implementation details; in this section we have used few contexts over which we have provided certain functionalities over which users can give their preferences. This forms the base for the application of Genetic Algorithm which we propose as an approach to achieve dynamic reconfiguration of context-aware middleware system. Section IV points out future directions of research in the area of context-aware middleware for mobile computing. In this section we have given brief description of Genetic Algorithm and Section V concludes the paper.

II. RELATED WORK

Philip Moore, Mike Jackson and Bin Hu, has given the concept of constraint satisfaction [2]. Their research found that, in the context-aware systems there is a general requirement to accommodate constraints and preferences and generally, with increased complexity, the issue of constraint and preference compliance with the avoidance of anomalies in the processing of contextual information has become increasingly important and challenging. Violation of constraints and preferences must be mitigated to avoid negative reactions to context-aware systems and to promote a positive response. The processing of contextual information in an intelligent context aware system offers the potential to personalize service provision to meet user's needs, desires, and interests. Technological developments, the interest in personalization, and the related increased complexity make constraint violation more probable and thus mitigation more important. Mitigating constraint violations is therefore a critical element in intelligent context aware systems.

A. Constraint Satisfaction

Constraint satisfaction relates to the ability of a context-aware system to accommodate system defined constraints, expressed user preferences, and security (generally) defined in the form of policy constraints defined as access rights and permissions. Examples of system constraints include restrictions based on the context for achieving better utilisation of scarce resources. Examples of user-defined constraints include restricting service provision to specified locations and times; user-defined preferences reflecting a user's predefined Beliefs Desires and Interests (BDI).

The concept of Constraint Satisfaction and the issue related to the Balance of user control [6] forms the base of our work. Context-aware applications may not always adapt as the user expects, and may cause users to feel loss of control over the behavior of their applications. In traditional applications, the trade-off between user control and software autonomy has been fixed at design-time. In contrast, context-aware

applications need to adjust the balance of user control and software autonomy at run-time. Thus, in our context-aware middleware system we proposed to use the new approach of genetic algorithm, a mechanisms to strike a suitable balance between user control and software autonomy.

Research work of Charith Perera, Arkady Zaslavsky, Peter Christen and Dimitrios Georgakopoulos has given the detail knowledge about the context life cycle [3]. We have referred their work to gain better understanding of the methods and techniques involved in different phases of the context life cycle. In the next section we present the partial implementation of our work and the methods or techniques used for data acquisition and processing of contextual information.

III. IMPLEMENTATION

Context awareness involves performing data acquisition from sensors, context recognition and other tasks necessary to complete before the context can actually be used. Assigning the data acquisition and context processing tasks to applications makes them almost impossible to reuse. One solution to such a problem is to decouple the tasks from applications and move desired functionality to the lower layers. Such layers, which serve the needs of applications, usually form a special layer called middleware. This middleware layer hides the heterogeneity and distributed nature of devices measuring the context information. Our context-aware middleware serves the context needs of applications. "Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". [6] We have focused on the contexts that can enhance the functionality of mobile computing devices, like battery power, storage memory and temperature of the device, also the contexts like location of the user, noise intensity around user.

Here we present the work flow of our middleware system and the stepwise brief description of methods or techniques used to implement the phases mentioned in the flow diagram below-

A. Context Acquisition

There are three ways to acquire context: sense, derive and manually provided.

Sense: The data is sensed through sensors, including the sensed data stored in databases (e.g. retrieve temperature from a sensor, retrieve appointments details from a calendar).

Derive: The information is generated by performing computational operations on sensor data. These operations could be as simple as web service calls or as complex as mathematical functions run over sensed data (e.g. calculate distance between two sensors using GPS coordinates).

Manually provided: Users provide context information manually via predefined settings options such as preferences (e.g. understand that user doesn't like to receive event notifications between 10pm to 6.00am).

Most of the devices we use today are equipped with a variety of sensor. The data retrieved from physical sensors is called low-level context. We have used these sensors for data acquisition of different context values. The continuous sensing of data helps in dynamic reconfiguration of services provided by our middleware. We have used Sound Sensor to sense the noise intensity and based upon its preference value pre-set by the user, we perform operations to provide service to the user and in same way other sensors like Power Sensor, Location Sensor, Storage Sensor and Temperature Sensor are used to provide respective services related to them.

B. Context Modelling

We used Logic Based Modelling technique where facts, expressions, and rules are used to represent information about the context. Rules are primarily used to express policies, constraints, and preferences. It provides much more expressive richness therefore; reasoning is possible up to a certain level. The specific structures and languages that can be used to model context using rules are varied. However, lack of standardization reduces the re-usability and applicability. Furthermore, highly sophisticated and interactive graphical techniques can be employed to develop logic based or rule based representations. As a result, even non-technical users can add rules and logic to the systems during run time. Logic based modelling allows new high-level context information to be extracted using low-level context.

C. Context Reasoning Decision Models

Context reasoning can be defined as a method of deducing new knowledge, and understanding better, based on the available context. It can also be explained as a process of giving high-level context deductions from a set of contexts. The requirement of reasoning also emerged due to two characteristics of raw context: imperfection (i.e. unknown, ambiguous, imprecise, or erroneous) and uncertainty.

Rules: This is the simplest and most straightforward methods of reasoning. Rules are usually structure in an IF-THEN-ELSE format. It allows the generation of high level context information using low level context. Most of the user preferences are encoded using rules. Rules are also used in event detection.

D. Context Distribution

Context distribution is a straight forward task. It provides methods to deliver context to the consumers. From the consumer perspective this task can be called context acquisition. There are two methods to that are often used in context distribution:

Query: Context consumer makes a request in terms of a query, so the context management system can use that query to produce results.

Subscription (also called publish / subscribe): Context consumer can be allowed to subscribe with a context management system by describing the requirements. The system will then return the results periodically or when an event occurs (threshold violation). In other terms, consumers can subscribe for a specific sensor or to an event.

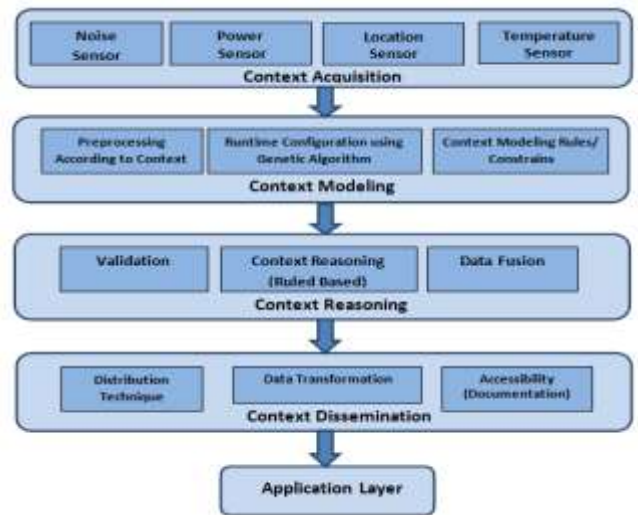
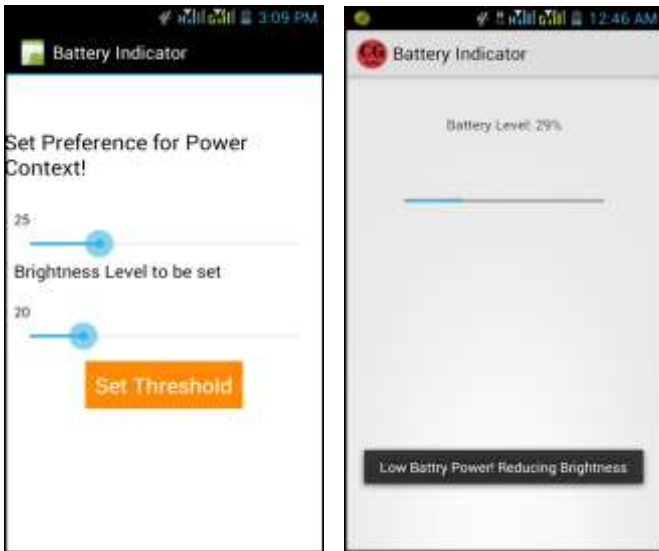


Fig. 1 Context life cycle

We have identified battery power, storage memory, temperature of the device, location of the user also the noise around the user as important context. Here we discuss some of the services provided by our middleware system over this context to enhance the functionality of the mobile computing devices-

A. Power Context

Battery power is the whole and sole of every mobile computing device thus this resource need to be used very efficiently. We used the power sensor to sense the battery level; it continuously senses the remaining battery of the device. As we are using rule based method for context modeling the user can set some threshold value and upon detecting the cross-over of pre-set threshold value our middleware perform various operations. Such as based on the primary context information we provide simple functionality of reducing brightness which does not affect the user but it definitely reduces the battery consumption and enhances the working of computing devices. Using this primary context information we can deduce the secondary context information which can be used find applications that are consuming more battery power and such applications can be managed based on the priorities set by the user. By performing data fusion of information obtained from battery sensor and temperature sensor we can further provide functionality for reducing heating of mobile devices.



B. Noise Context

We have used Sound Sensor so that we can continuously sense the noise around user. The user can pre-set any value as threshold and if the noise around user goes above the threshold value then the action will be performed to alert the user. The user can set any sound clip or can choose to vibrate his device if the noise exceeds the threshold value. It helps user if the user is unaware of high intensity sound buzzing around him, particularly it happens in the cases when the user is unable to hear properly or user has put in the ear phones then to make him alert about the noise around him we vibrate the device. For instance, if a user is listening songs on his handset while walking along a road and a vehicle is approaching him blowing horns to move him aside. Our middleware provide service in such type of situations by vibrating the mobile to alert the user.

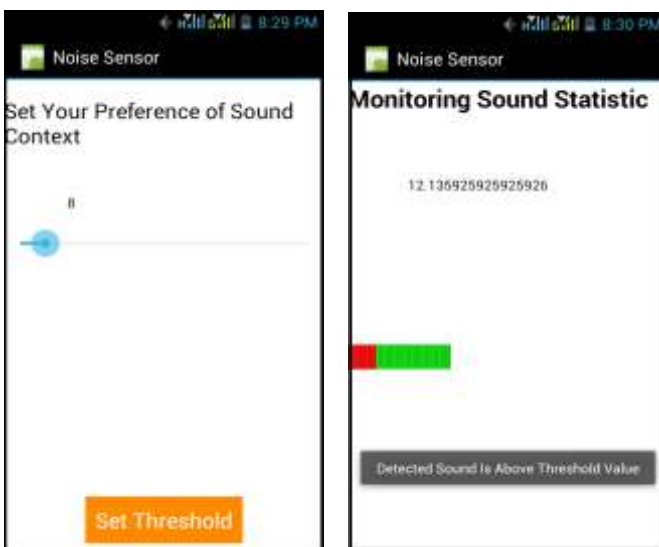


Fig. 3 Implementation over noise context

C. Location Context

Location of the user plays an important role in deciding the computing environment for the applications. Based on the location context we can infer much information related to user. This helps in providing better understanding for service provisioning to the users. We have used GPS to find the latitude and longitude position of the user, the context has been used to let user save his location. The user can save the profile preference based on the location. This service help user to automatically change the profile to vibration or ringer mode as per his given preference when he enters the proximity to the saved location. This service free user from every time changing his profile while entering the places like hospitals where he is required to keep his mobile to vibration mode and remembering to switch his profile to ringer mode when he leaves such places.

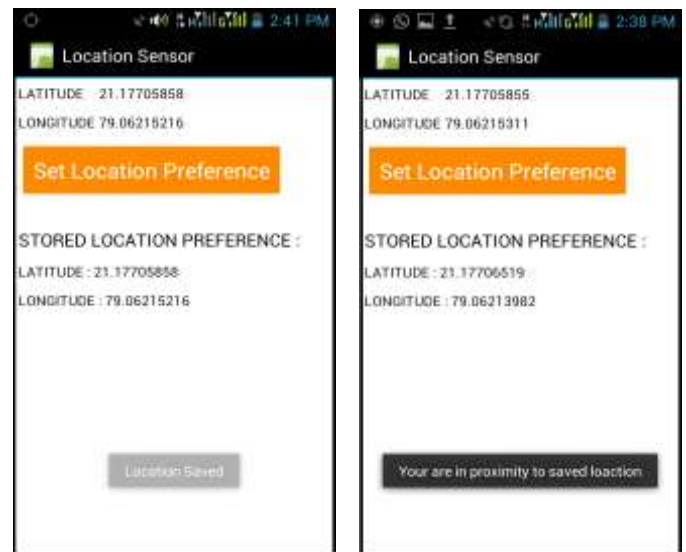


Fig. 4 Implementation over location context

IV. PROPOSED WORK

Genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and recombination.

The genetic algorithm is a population-based iterative optimization method. Instead of just pushing iteratively a single candidate toward the optimal solution, it acts on a set of such solution. The evolution usually starts from a population of randomly generated individuals and already present in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified to form a new population. The new population is then used in the next iteration of the algorithm.

A typical genetic algorithm requires two things to be defined: a genetic representation of the solution domain, and a fitness function to evaluate the solution domain. Genetic

algorithms are inspired by Darwin's theory about evolution – the "survival of the fittest". In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

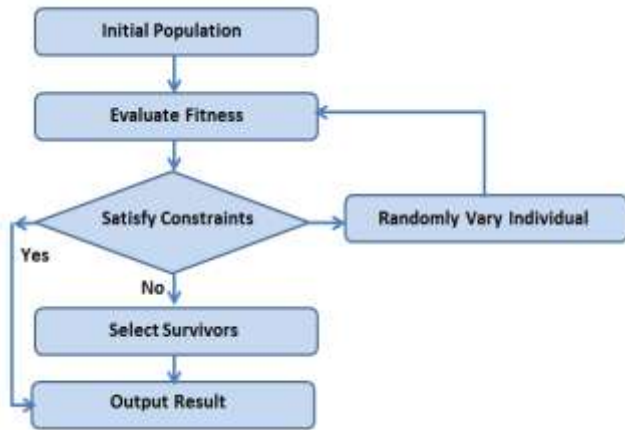


Fig. 5 Flow of Genetic Algorithm

In our work we have created a database to store the current value of different contexts and the preferences selected by the users over these contexts. This creates the initial population for the implementation of Genetic Algorithm. The algorithm evaluates fitness over each of these preferences and generates the survivors based on the system defined constraint satisfaction. In this way it generates an optimal solution in terms of best preference for the users.

V. CONCLUSIONS

In this paper we have presented different services based on the context awareness that can be provided to the user. We propose Genetic Algorithm as the new approach for the provision of dynamic reconfiguration to context-aware middleware, so that it can adapt and reconfigure itself to enhance the behaviour of the applications. This will help

designers building mobile applications to better utilise the scarce resources of mobile device like CPU power, battery life and memory. In our future work we propose to use the concept of constraint satisfaction with the Genetic Algorithm to find optimized solution. The main focus of our middleware is to support personalization and adaptation based on context-awareness.

REFERENCES

- [1] Adane D. S., Sathe S. R. and Moghe C. S., "Current Status Of Middleware Platforms For Mobile Computing" Proceedings of the IEEE International conference on Wireless Networking and Mobile Computing" (ICWNMC'05) held at Velammal Engineering College, Chennai from December 28-30
- [2] Philip Moore, Mike Jackson, Bin Hu, "Constraint Satisfaction in Intelligent Context-Aware Systems" 2010 International Conference on Complex, Intelligent and Software Intensive Systems
- [3] Charith Perera, Arkady Zaslavsky, Peter Christen and Dimitrios Georgakopoulos "Context Aware Computing for The Internet of Things: A Survey" IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. X, NO. X, XXXX XXXX, 5 May 2013
- [4] Beresford A. R., Stajano F., "Location Privacy in Pervasive Computing", *Pervasive Computing*, IEEE, Volume 2, Issue 1, Jan-Mar2003,pp.46-55.
- [5] Campbell R., Al-Muhtadi J., Naldurg P., Sampemane G., and Dennis Mickunas M., "Towards Security and Privacy for Pervasive Computing" ISSS'02 Proceedings of the 2002 Mext-NSF-JSPS international conference on Software security: theories and systems, pages 1-15.
- [6] Marco B. and Leonardo B., "A survey about context-aware middleware", Retrieved from <http://www.slideshare.net/LeoBruni/a-survey-about-contextaware-middleware>, June 16, 2009.
- [7] Abowd G. D., Dey A. K., Brown P. J., Davies N., Smith M., and Steggles P., "Towards a better understanding of context and context-awareness," in Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, ser. HUC '99.London, UK: Springer-Verlag, 1999, pp. 304-307. [Online]. Available: <http://dl.acm.org/citation.cfm?id=647985.743843>
- [8] Srinivasu Yelsuri , G.B.V Padmanadh. "A Novel Architecture for Dynamic Invocation of Web services from Mobile". International Journal of Engineering Trends and Technology (IJETT). V4(10):4404-4407 Oct 2013.ISSN:2231-5381. www.ijettjournal.org