

Towards Proactive Context-Aware Service Selection in the Geographically Distributed Remote Patient Monitoring System

Pravin Pawar, Bert-Jan van Beijnum, Hailiang Mei, Hermie Hermens

¹Department of Electrical Engineering, Mathematics and Computer Science

University of Twente, The Netherlands.

{p.pawar, Beijnum, h.mei, h.hermens}@cs.utwente.nl

Abstract

In the mobile (M)-Health domain, the Remote Patient Monitoring System (RPMS) facilitates continuous collection, transmission and viewing of the patient vital signs data. Furthermore, in case of an emergency it provides context-aware Emergency Response Services (ERSs) such as the doctor, paramedic, ambulance and hospital to the patient. In the existing RPMS, the components responsible for the discovery and selection of ERSs are centralized; and the ERS are selected and invoked reactively; i.e. on the occurrence of an emergency. However, in the distributed RPMS, such reactive approach could take a significant amount of time. If ERSs could be discovered and selected proactively before an emergency occurs, it could save valuable time in dealing with the patient emergency. However, such proactive approach requires more resources as compared to the reactive approach. Herein we present a logical architecture for the distributed RPMS, investigate the issues involved in the proactive context-aware ERSs selection and the simulation methodology to analyze resource utilization and emergency response time tradeoff of the proactive vs. reactive approaches.

1. Introduction

The context-aware Remote Patient Monitoring System (RPMS) developed as a part of the AWARENESS project [1] facilitates the following:

- Real-time collection of the patient vital signs (e.g. ECG, EEG) by means of a Body Area Network [2];
- Real-time transmission of the vital signs using the wireless connectivity to the healthcare professionals;
- Seamless handover over different wireless communication technologies such as WLAN, GPRS and UMTS;

- Context-aware infrastructure to sense the context (e.g. location, availability, activity, role) of the patients and Emergency Response Services (ERSs) to provide assistance to the patient in case of an emergency. An ERS could be *fixed* (e.g. hospital) or *mobile* (e.g. caregiver). A mobile ERS is published in the RPMS using the concept of the so-called *Nomadic Mobile Service* as described in [3]. Using this concept, the service hosted on the mobile device participates in the service discovery network by means of a proxy called as the *Surrogate Host*. For details, we refer to [3].

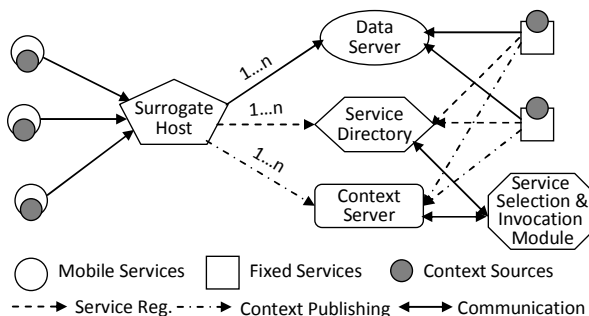


Figure 1: Logical Architecture of Centralized Remote Patient Monitoring System

Based on the concepts of the *Service Oriented Architecture* (SOA) [4], each ERS registers itself in the service directory. Moreover, each ERS has an associated *Context Source* which publishes the service's context to the context server. The context-aware *Service Selection and Invocation Module* (SSIM) is responsible for searching the suitable services (e.g. the nearest caregiver) in case of an emergency. The SSIM receives the emergency situation context information and the ERS is selected according to a certain set of *Service Selection Identifiers* (SSID), such as *nearest*, and/or *available*. The dynamic context information (e.g. current location and availability) required by SSIM is provided by the ERS context

source via the *Context Server*. The *Data Server* is responsible for storing static context; e.g. patient profile. Figure 1 shows the logical architecture of existing RPMS which incorporates centralized components to take on a role of the surrogate host, service directory, context server and the SSIM. In the centralized RPMS, the ERSs are discovered and invoked *reactively*; i.e. on the occurrence of an emergency. Due to the centralized components, the computation and communication overhead for these operations is not significant [5].

However, in the pervasive computing environment, the centralized solutions do not scale up well [6]. Hence, to be able to serve larger geographic areas and higher number of patients, a distributed solution is necessary. Herewith, we propose a logical architecture for the distributed RPMS where multiple RPMS components are distributed according to the geographic zones. Due to the distributed nature of services and other RPMS components therein, the reactive ERSs selection approach could take significantly higher amount of time to discover and select ERSs adding to the emergency response time. This could be a potential bottleneck in dealing with the patient's emergency situation. One of the solutions to this problem is to select a set of ERSs *proactively* before an emergency occurs and invoke already selected services immediately. However, because of the user mobility and other context changes (e.g. Caregiver becomes busy), the proactive ERSs selection needs to be recomputed from time to time. Hence, this approach requires higher computational and communication resources as compared to the reactive approach. To analyze the overhead involved and the time-savings benefits of the proactive approach, herein we propose a simulation methodology for comparing these two approaches. The proposed methodology is motivated from the event-based simulation approach in [7] and considers the zonal structure of Singapore.

In summary, our contribution reported in this work is as follows: 1) Logical architecture for the geographically distributed RPMS; 2) Context-aware proactive ERSs selection approach; and 3) Simulation methodology to evaluate the resource utilization and time savings tradeoffs of the proactive approach vs. reactive approach.

The above three contributions are reported in the Section 3, Section 4 and Section 5 of the paper respectively. Section 2 of the paper presents related work. Section 6 concludes the paper and elaborates the future work.

2. Related Work

The architecture of the distributed RPMS presented in this paper uses a number of concepts proposed in the related work. Leonhardi et. al. [8] presents architecture for the large-scale location service which aims at managing dynamic location information for a large number of mobile objects. The *location server* associated with each zone keeps the location updates in the main memory, while registration information is stored in the traditional database. Storing updates in the main memory provides faster response to the location queries. Grossman et. al. [6] claims that in the large-scale scenarios, it is beneficial to have special-purpose servers that are optimized for managing a certain class of context data; such as *static context* (e.g. map of a building) and *dynamic context* (e.g. location). Drosdol et. al. [9] extends the argument presented in [6] to explain the issues involved in managing a large number of mobile objects (so called bulky elephants). Moreover, [9] proposes a number of architectures for the large-scale system where the specialized index servers are capable of resolving spatial, object ID based or mixed queries efficiently.

Regarding proactive context-aware service selection approach, we did not find any directly related work. However, Hesselman et. al. [10] present an approach for the persistent context-aware service selection; where the request for context-aware service selection is kept persistent in the memory; services context and client context are processed in real-time to recommend best matching service to the client based on the *service selection identifier* (e.g. nearest). We use this approach as an enabler for the proactive service selection. The performance evaluation of the context-aware service discovery approach conducted in [10] showed that in the centralized system, the difference between the runtime performance of the regular service selection (without considering services and client context) and the context-aware service selection is relatively minor; especially when we compare the waiting time for discovering services. However, there is an additional effort involved in the creation of the context sources. We try to build up on the experience in [5] while comparing the performance of the reactive and proactive approaches described herewith.

Though we utilize several concepts proposed in the related work in the distributed RPMS architecture, we distinguish our work in the following aspects: 1) We consider a specialized m-health domain application problem for the remote patient monitoring; 2) We introduce additional components such as fixed/mobile services, service directory, surrogate host and SSIM in

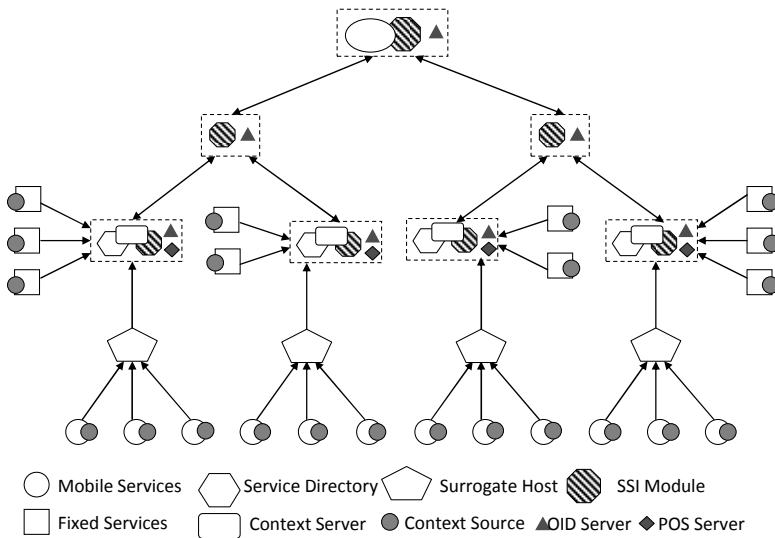


Figure 2: Logical Hierarchical Architecture of the Distributed RPMS

this work as compared to [8] and [9]; 3) We don't apply the index server components proposed in [9] in their original form, but modify/extend them to our application needs. Moreover, the proactive context-aware service selection approach and related algorithms is an entirely new contribution in this work.

3. Architecture of the Distributed Remote Patient Monitoring System

In the RPMS, we identify the following objects and services published by them. Every object has an associated unique *Object Identifier* (OID) and geographic *POSition* (POS) in terms of GPS coordinates.

- 1) *Patient*: In the traditional sense, a patient is the user of a *Monitoring Service*. However, since this service is activated and published by the mobile device being carried by the patient, we say that the patient publishes monitoring service in the RPMS.
- 2) *Caregiver*: The caregiver publishes a mobile ERS called as the *Caregiver Service*. In case of an emergency, it is desired that the caregiver rush to the patient's location and provide necessary pre-hospital assistance.
- 3) *Ambulance*: The ambulance publishes a mobile ERS called as the *Ambulance Service*. An ambulance provides transportation to the hospital and pre-hospital care to the patient.
- 4) *Hospital*: The hospital publishes a fixed ERS called as *Hospital Service* and it provides hospital treatment to the patient after the emergency occurs.

- 5) *Doctor*: The doctor publishes a fixed ERS called as *Doctor Service*. The doctor is responsible for monitoring the patient's vital sign data and instructing the invocation of appropriate ERSs depending on the nature of emergency.

For selecting all the suitable ERSs, SSIM applies *nearest* and *available* SSIDs. Please, note that this set of SSIDs could be extended further (e.g. *fastest*) provided that the required context information (e.g. speed, mean of transport) is provided by the ERS context sources.

In the geographically distributed RPMS (Figure 2) presented herein, a geographic area under consideration is sub-divided into several zones and

every zone has an associated service directory, context server, and SSIM. In addition, as motivated from [9] each zone has an associated *OID-index server* and *POS-Server* (as explained in the following) for efficient querying.

Table 1: Structure of OID-index Record

OID	Type	POS-Server	Context Server	Service Directory
HP12	Fixed	PS12	CS12	SD12
CG12	Mobile	*->OI12	*->OI12	*->OI12

OID-index server: The OID-index server is aimed at the efficient processing of OID queries (e.g. *Which context server stores the context information of caregiver object with OID CG12?*) by keeping track of the object data (such as context information and provided service) storage location of each object. Since we provision different servers for these purposes, OID-index points to multiple servers. The OID-index space is hierarchical, i.e. there is a hierarchy of OID-index servers similar to that of the location servers in [8]. Because of the involved mobility, to reduce frequent updates, for the mobile objects, the higher level OID-index server maintains a pointer to the lower level OID-index server. The structure of OID-index record is shown in the Table 1.

POS-server: There is one POS-server in every zone and it is responsible for retrieving objects based on their positions. It maintains only minimum amount of data: the OIDs and their current positions. In addition to this, the POS-Server also stores the *zone adjacency list* (the zone IDs which are immediate neighbors) for

the geographic zone it is responsible for. This list is specially created because of the reason that potentially there is a possibility that the nearest caregiver and ambulance are located in the zone adjacent to that of the mobile patient. The POS-server is aimed at the processing of spatial queries (e.g. *What are the OIDs of the caregivers located in the zone Z12?*)

Using a combination of these two servers, it is possible to answer an arbitrary context-aware query such as *Find the nearest caregiver to the patient with OID PT12.* (Refer to the Section 4 for details). The distributed RPMS has a hierarchical architecture as described in the following:

Level 0 (leaf level): The leaf level consists of the fixed ERSs, mobile ERSs, monitoring service and their associated context sources. For the mobile services, the leaf level node also includes their corresponding surrogate host, because as conceptualized in [3], the client of the mobile service is transparent to the fact that the service is hosted on the mobile device and interacts with the surrogate host as the service provider. The leaf level nodes register with the service directory and context server in their designated geographic area.

Level 1: A level 1 node consists of a service directory, context server, level 1 SSIM, POS server and level 1 OID-index server. Conceptually, the functions of these entities are decoupled from each other, however physically; they could be hosted on the same machine.

Level 2 upwards excluding root level: These levels consist of the SSIMs, and OID-index servers. The SSIM at these levels communicate with the service directory and context server at the level 1 for obtaining the required information. The total number of levels depends on the number of zones, number of objects, concurrent/maximum number of operations in the system, the computational capacity of basic RMPS components and the capacity of the communication links.

Root level: The root level consists of the root level OID-index, SSIM and the data server. The data server is a centralized component and is responsible for storing static context data such as patient's profile.

3.1. Services Data and Context Flow

In the description of this section, we assume that the preliminary steps such as objects registration has been completed, an object is assigned unique OID and the object possesses necessary software and hardware to publish services and the context information in the distributed RPMS. We also assume that the basic RPMS components such as POS-Servers, OID index servers, service directories and context servers are

aware of the existence of each other as well as the geographical zones they are serving to, wherever applicable.

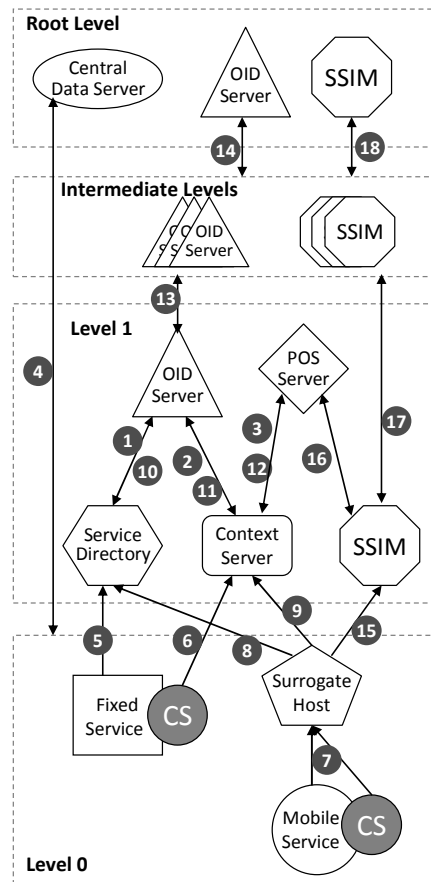


Figure 3: Services Data and Context Flow in the Distributed RPMS

The services data and context flow in the distributed RPMS is shown in the Figure 3. Every level 1 OID-index server subscribes to the service directory (*interaction 1 in the Figure 3*) and context server (*interaction 2*) in its zone to receive the notifications about service and context source registration/deregistration events respectively. In addition, a POS server subscribes to the context server (*interaction 3*) to receive the object location changes events. When a fixed or mobile service is activated, it first contacts the central data server (*interaction 4*) to obtain the *Universal Resource Identifier* (URI) of service directory, context server and surrogate host (only in the case of mobile service) in its geographic zone. On receiving the requested information, the fixed service registers itself and its context source in the service directory (*interaction 5*) and in the context server (*interaction 6*) respectively. In contrast, the

mobile service initializes communication with the surrogate host (*interaction 7*), which takes care of registering the mobile service (*interaction 8*) and its context source (*interaction 9*).

Once a new service (context source) is registered, the service directory (context server) sends notification to the OID-index server (*interaction 10 (11)*). Also, the context server sends notification to the POS-server (*interaction 12*). The OID index server updates its records and this update is propagated to the root level OID index server (*interaction 13 - 14*) as described in the Section 3.

For the proactive ERSs selection approach, if the newly activated service is a patient’s monitoring service, the surrogate host sends ERSs selection request to the level 1 SSIM (*interaction 15*). On receiving this request, the SSIM contacts the POS server (*interaction 16*) to obtain the zone adjacency list. On receiving this list, the SSIM determines the *least common SSIM* to the current zone and adjacent zones. The least common SSIM is the first SSIM which is reachable by all these zones while traversing hierarchy towards the root. The ERSs selection request is later forwarded to the least common SSIM (*interaction 17*) which could eventually reach root level SSIM (*interaction 18*) in case the required ERSs are not found in these zones.

In case of the reactive ERSs selection approach, similar sequence of steps is followed, albeit after the occurrence of an emergency. Further steps in the proactive and reactive ERSs selection mechanisms are described in the Section 4.

3.2. Handling Mobile Object’s Geographic Mobility

Because of the mobility, the mobile objects may cross the boundaries of the geographic zones served by a particular surrogate host. This requires that a handover to the appropriate surrogate host needs to be done. In our case, the handover decision is taken by the surrogate host serving the mobile object, because the surrogate host is aware of the existence of other surrogate hosts, their corresponding geographic boundaries and the current location of the mobile object. Analogous to the vertical handover process described in [11], we explain various phases involved in the surrogate host handover in the following:

Handover information gathering phase: This phase collects information required for both; to identify the need for handover and subsequently initiate it. The surrogate host is constantly monitoring the location information received from the mobile service’s context source to determine if the mobile object has moved outside the geographical zone. If this holds true, to

avoid so called ping-pong effect, based on the technique used in [12] the surrogate host initializes a dwell timer for the corresponding mobile object. The dwell timer has a certain value (e.g. 1 minute, which is the time required for the mobile object to move 1 km at the speed of 60 kmph). If after the dwell timer expiry, the condition remains same i.e. the mobile device is still out of the geographic zone, then the handover decision phase is initiated. This is a possible algorithm for handover initialization, but other algorithms (for example those that make use of tracking [13] by estimation of velocity etc.) are possible within this framework.

Handover decision (surrogate host selection) phase: This phase determines the surrogate host which is serving the new geographic zone the mobile object is in.

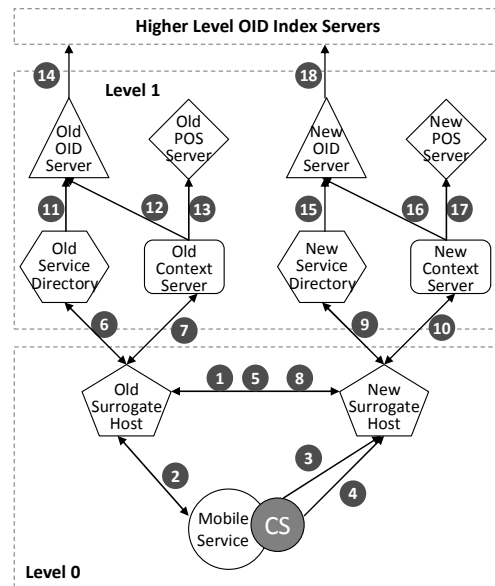


Figure 4: Interactions during the Surrogate Host Handover

Handover execution phase: This phase performs the actual handover to the surrogate host selected in the second phase. The events in this phase are shown in the Figure 4. The old SH sends the new SH OID of the mobile object, the service description and context source description and instruct the new SH to takeover the mobile service (*interaction 1 in the Figure 4*). On receiving the acknowledgement from the new SH, the old SH instructs the mobile object (*interaction 2*) to send further service and context data to the new SH, which is done subsequently (*interaction 3 and 4*). The new SH confirms to the old SH (*interaction 5*) that it is receiving data from the mobile object. Based on this event, the old SH sends the *service (context source)*

removal due to handover message to the old service directory (old context server) (*interaction 6 (7)*) and instructs the new SH to register the mobile service (context source) in the new service directory (new context server) (*interaction 8*); which is later done subsequently (*interaction 9 and 10 respectively*). On the removal of the service (the context source) from the old service directory (old context server), the old OID-index server receives the service removal (context source removal) notification (*interaction 11(12)*). Similarly, the old context server notifies same to the old POS server (*interaction 13*). In contrast, the new OID-index server receives the service (context source) registration event (*interaction 15 (16)*). As well as the new POS server receives the context source registration event (*interaction 17*). The updates in the OID index servers are propagated upwards (*interaction 14 and 18*) and appropriate updates are done till the least common OID index server of the old and new geographic zones. Please, note that the handover of mobile objects also has consequences on the SSIM functioning. These are explained in the Section 4.

4. Proactive Context-Aware Services Selection and Invocation Mechanism

Summarizing the discussion in the Section 3, the initial ERSs selection request is handled by the least common SSIM (so called *initial SSIM*) to the zone of requesting mobile device and its adjacent zones. An illustrative example showing a hierarchical structure of SSIMs is shown in the Figure 5. In this figure, the level 1 zones and their SSIMs have same label. The requester mobile device is in the zone G. The zone adjacency list of zone G is {E, F, H, I}. The request for initial selection of ERSs is forwarded by the SSIM G to SSIM Q which is the least common SSIM to the zones E, F, G, H and I. On receiving this request, the SSIM Q executes the following steps to make a primary selection of ERSs:

Step 1: Obtain a list of the objects providing ERSs from the POS servers of the requestor zone and its adjacent zones. E.g., in the Figure 5, SSIM Q contacts POS Servers of zones SSIM E, F, G, H and I.

Step 2: Contact the OID-index server to obtain a list of context servers and service directories of the objects obtained from the POS server.

Step 3: Use the ERSs services and context information to select the nearest and available caregiver, ambulance and hospital services using a context-aware service discovery approach proposed in [10].

Step 4: If any of the suitable ERSs could not be found in the requestor zone and its adjacent zones, then the ERSs selection request is forwarded to the least

common SSIM of the zones adjacent to the zones in the zone adjacency list, where step 1 and step 2 are executed again. E.g., in the Figure 5, the request is forwarded to SSIM R.

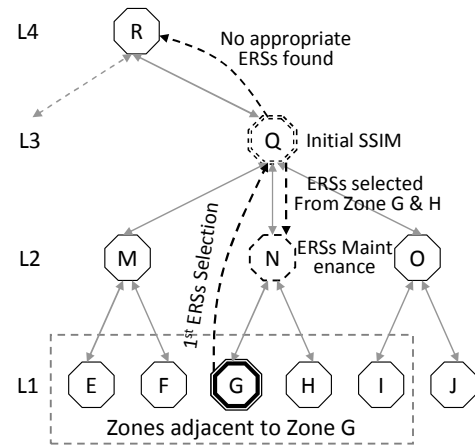


Figure 5: Example Showing Initial Selection of ERSs and ERSs Maintenance Phases

Step 5: To avoid unnecessary propagation of ERSs context information, the set of selected ERSs is further sent to the least common SSIM (so called *maintaining SSIM*) serving the zones of the selected ERSs for its maintenance. E.g., in the Figure 5, the suitable ERSs are in the zones G and H. Hence, the set of selected ERSs is sent to SSIM N for the maintenance.

Step 6: The maintaining SSIM subscribes to the service directories and context servers of the ERSs for the ERSs service and context changes as follows:

- 1) *Caregiver status changes to busy.*
- 2) *Doctor's status changes to busy.*
- 3) *Hospital is full and can not accommodate further number of patients.*
- 4) *The relative distance between of the patient and caregiver (or between the patient and ambulance) exceeds certain value. (Here also the concept of dwell timer is applied to reduce unnecessary re-selection of the ERSs).*
- 5) *Any of the ERSs (caregiver, ambulance and hospital) goes offline.*
- 6) *The caregiver (or ambulance) moves out of the current geographic zone.*

On these changes, the maintaining SSIM request initial SSIM to re-select only the ERS the service and context change of which has resulted in one of the above conditions.

Handling patient mobility across zones: If the patient moves out of the current geographic zone, the current ERSs selection is dropped and the proactive context-

aware service selection mechanism is re-executed, because the new geographic zone could have different adjacent zones than that of the old geographic zone.

A flowchart comparing the proactive and reactive ERSs selection and invocation approaches is shown in the Figure 6. In the proactive approach, on the occurrence of an emergency, the doctor service is invoked (i.e. a doctor is able to see patient's vital signs and take further decision), other ERSs are alerted by providing the patient's ID and location information and based on the doctor's decision further actions are taken. In contrast, in the reactive approach, first the doctor service is selected and later it is invoked. Based on the doctor's decision, further ERSs selection and invocation is performed. The ERSs selection in the reactive approach is done in a way similar to that of the proactive approach. In both of these approaches, SSIM subscribes to the context source of the monitoring service to get notified whenever an emergency occurs.

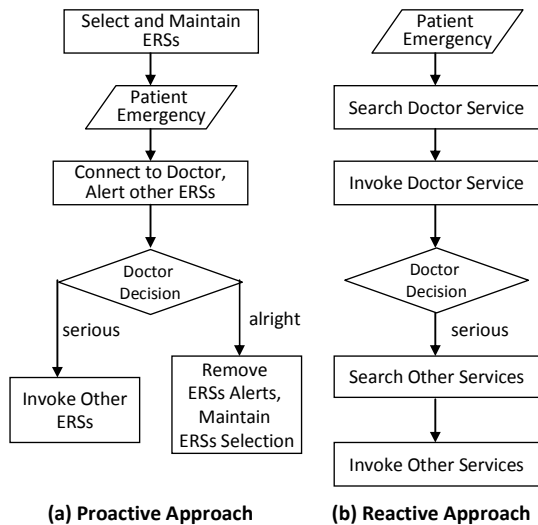


Figure 6: Flowchart of Proactive and Reactive ERSs Selection and Invocation Approach

5. Simulation Methodology for the Performance Evaluation

Because of the resource and time limitations, it is not possible to build the distributed RPMS system and perform real experiments. Hence, we are building an event based RPMS simulation system inspired by [7]. In the distributed RPMS system, there are a number of events happening concurrently. Each event has an associated timestamp, ID, name, sender ID, receiver ID, and other event-specific information necessary for processing the event. Processing certain event could result in other number of events. The idea here is to

order and process the events according to their timestamp (temporal ordering), process events one by one and queue the possible resulting events in the temporal order for further processing.

Since the fundamental idea in this work is the distribution of the system components according to the geographic zones, for the simulation to be realistic, it is required to consider a particular geographic topology. In today's Internet, fairly accurate geographic information is available. For example, *Google maps server* provides accurate location maps. In this work, we use the geographic topology of Singapore as a reference. The geographic zones are considered according to the postal code regions. Similar to [14], we follow the approach of modeling user movements between the source and destination coordinates along the route obtained from the Google maps server.

We target the following four population classes which could benefit from the remote patient monitoring system: 1) elderly, 2) epilepsy patients, 3) patients with chronic pain and 4) acute pregnancy cases. We are working with the medical professionals in Singapore to obtain various statistics concerning the following:

- 1) Zone-wise distribution and mobility behavior of the target population;
- 2) Potential number of caregivers, doctors, ambulances, their zone-wise distribution, mobility behavior and availability pattern;
- 3) Number of hospitals, their zone-wise distribution, number of intakes for the remotely monitored patients;
- 4) Time of day wise emergency occurrence rate for the target population;
- 5) Patient arrival rate and average treatment time for every class of patient.

Apart from the above, we are also working on the simulation of the computation and communication characteristics of the distributed RPMS. Some of them include the creation of the objects representing distributed RPMS components, assigning computational resources to the server objects and assigning QoS characteristics to the wired and wireless communication links in the distributed RPMS.

6. Conclusion and Future Work

In this work, we claim that in the healthcare domain, the proactive context-aware *Emergency Response Services (ERSs)* selection approach which selects the ERSs before an emergency occurs could be beneficial in the distributed *Remote Patient Monitoring System (RPMS)* to reduce the emergency response time.

Towards analyzing this argument, we have outlined a logical architecture for the distributed RPMS where the RPMS components are distributed geographically. In line with the previous architectures for handling large scale mobile objects and their context information, the proposed architecture consists of a hierarchy of RPMS components adapted and extended to the application problem under consideration. We describe in details the ERSs data and context flow, mobility handling mechanisms and proactive ERSs selection approach in such architecture. Further, we outline the simulation methodology for the performance evaluation of the proactive vs. reactive approaches to analyze their tradeoff for the resource utilization and emergency response time as a result of both of these approaches. We are currently working on gathering real-life data from the healthcare professionals to make simulation as realistic as possible. In the future, we will be working on the event based simulation of the distributed RPMS modeled according to the geographical zones and patient population distribution in the Singapore region. The simulation will specifically analyze the resource utilization (e.g. computational and communication requirements, time spent for the service selection and invocation operations) and emergency response time savings of the proactive vs. reactive ERSs selection approaches.

7. Acknowledgements

The authors acknowledge support of Dr. Jit Biswas of the Institute for Infocomm Research, Singapore in the preparation of this manuscript. This work is supported by Dutch Freeband Awareness project (under grant BSIK5902390, <http://awareness.freeband.nl/>) and overseas student attachment grant from the Institute for Infocomm Research, Singapore.

References

- [1] Wegdam, M., "AWARENESS: A project on Context AWARE mobile NETworks and ServiceS", Proceedings of the 14th Mobile & Wireless Communications Summit 2005, 19-23 June 2005, Dresden, Germany.
- [2] van Halteren, A., Bults, R., et al., "Mobile Patient Monitoring: The MobiHealth System". *The Journal on Information Technology in Healthcare*, 2(5), 2004.
- [3] van Halteren, A. and P. Pawar. "Mobile Service Platform: A Middleware for Nomadic Mobile Service Provisioning", 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2006), June 2006, Montreal, Canada.
- [4] Papazoglou, M.P. and Georgakopoulos, D., "Service-Oriented Computing", *Communications of the ACM*, 46(10):25-28, October 2003.
- [5] Poortinga, R., Chapter on "Assessment of Context-Aware Service Discovery" in "Amigo overall middleware: Final prototype implementation & documentation", Deliverable 3.5, Amigo Project (www.hitech-projects.com/euprojects/amigo/), December 2007.
- [6] Grossmann, M., Bauer, M., Hönle, N., Käppeler, U. P., Nicklas, D., Schwarz, T., "Efficiently Managing Context Information for Large-scale Scenarios", 3rd IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom 2005), March 2005, Hawaii, USA.
- [7] Misra, J., "Distributed Discrete-Event Simulation", *ACM Computing Surveys*, Vol. 18, No. 1, pp 39-65, March 1986.
- [8] Leonhardi, A., Rothermel, K., "Architecture of a Large-scale Location Service", 22nd International Conference on Distributed Computing Systems (ICDCS'02), July 2002, Vienna, Australia.
- [9] Drosdol, T., Schwarz, T., Bauer, M., Großmann, M., Hönle, N., Nicklas, D., "Keeping Track of Flying Elephants: Challenges in Large-Scale Management of Complex Mobile Objects", *Tagungsband der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, 2004.
- [10] Hesselman, C., Tokmakoff, A., Pawar, P., Jacob, S., "Discovery and Composition of Services for Context-Aware Systems", 1st IEEE European Conference on Smart Sensing and Context (EuroSSC 2006), October 2006, Enschede, The Netherlands.
- [11] Kassar, M., Kevella, B., Pujolle, G., "An Overview of Vertical Handover Decision Strategies in Heterogeneous Wireless Networks", *Computer Communications* 31 pp. 2607–2620, 2008.
- [12] Hong, C. P., Kang, T. H., and Kim, S. D., "A Profile Based Vertical Handoff Scheme for Ubiquitous Computing Environment", *Asia-Pacific Network Operations and Management Symposium 2006*, September 2006, Busan, Korea.
- [13] Zhang, Z., Gao, X., Biswas, J., Wu, J. K., Moving Targets Detection and Localization in Passive Infrared Sensor Networks, 10th International Conference on Information Fusion (Fusion 2007), July 2007, Québec City, Canada.
- [14] Pawar, P., van Beijnum, B. J., Wac, K., Hermens, H., Konstantas, D., "Towards Location Based QoS-Aware Network Selection Mechanism for the Nomadic Mobile Services", 6th Annual IEEE Consumer Communications & Networking Conference, (IEEE CCNC 2009), January 2009, Las Vegas, Nevada.