

# A Software Development Model for Localization Systems

Warsun Najib, Martin Klepal, Widyawan, Dirk Pesch

*Centre for Adaptive Wireless Systems, Department of Electronic Engineering*

*Cork Institute of Technology,*

*Rossa Avenue, Bishopstown, Cork, Ireland*

warsun.najib@cit.ie, martin.klepal@cit.ie, widyawan@cit.ie, dirk.pesch@cit.ie

**Abstract**— Over the past few years various localization systems using a range location technologies have been developed. The need for a unified model for developing localization applications has arisen out of this work, a model which provides a unified abstraction of the current range of localization technologies. Out of this need, several localization frameworks have been proposed which provide a model for developing localization systems. However most of those have a centralized architecture, which limits their wider application. In this paper we present a software development model for localization systems based on a distributed and hierarchical architecture. Our model has a layered architecture, where each layer has interfaces to provide services to another layer. We develop an application as a prototype implementation of the framework to illustrate the advantages of our approach. The implementation of the prototype application has shown that our approach enables more flexible development of new localization applications with various different configurations than previous proposals.

## I. INTRODUCTION

Developments in the area of context aware computing have been growing rapidly over recent years, with researchers developing a wide range of context based applications. An important aspect of *context* is location, which is typically an output of a localization system. As part of this work, many localization systems have been developed for various applications, which also use other types of sensory information.

Currently, two localization frameworks have been proposed in the literature as a software development model. One of the most popular approaches is the Location Stack by the University of Washington [1]. The model is similar to the seven-layer Open System Interconnection (OSI) model for computer networks. Another model is the Universal Location Framework developed by Intel [2]. This framework consists of four layers: sensor layer, measurement layer, fusion layer, and application layer.

The functionality of localization systems has evolved and now requires more complex data processing and therefore the need for a more sophisticated software model for development of localization systems has become critical. Furthermore, many localization systems also need distributed data processing for scalability and performance reasons. In this paper we propose a software development model based on distributed, hierarchical network architecture. The model has a layered architecture, where each layer will provide an

Application Programming Interface (API). We have implemented a localization system for an emergency response scenario as a demonstration of the usefulness of the proposed model, which shows that our model enables more flexible design of new localization systems. Moreover, the model enables the implementation of effective data fusion of various location data such as measurements from different types of sensors such as wireless LAN, inertial sensors, and ultrasound sensors.

The remainder of the paper is organized as follow. In Section II the design principles are briefly described. The proposed localization framework is presented in the Section III. Section IV describes experiments conducted with a prototype implementation of the framework for our chosen scenario. Finally section V concludes the paper.

## II. DESIGN PRINCIPLES

The software development model is designed based on the following principles.

- *Unified software development model.* Availability of a common software development model for localization systems will enable the flexible implementation of various types of such systems.
- *Combination of location technologies.* We can improve estimation accuracy by fusing location data resulted from aggregation of more than one sensor technology. One example is the combination of measurement data of inertial sensors and ultrasound sensors in a fire-fighter scenario. We found that the combination will increase accuracy when the system tracks and guides the fire-fighters in the building.
- *Object localization is usually concerned with context and activities.* Location information is not directly used by the application itself. The reason for location estimation is to enhance context data, which is typically used by a higher abstraction level in user activities.
- *Distributed localization system.* Localization system architectures can be centralized as well as distributed. The proposed software development model will enable both approaches.
- *Scalability of localization system.* The proposed software development model should provide scalability

when needed by the localization application. The model should facilitate a common approach for developing both simple and complex localization systems.

### III. SOFTWARE DEVELOPMENT MODEL FOR LOCALIZATION SYSTEM

Based on the design principles described in the previous section we propose a localization framework consisting of seven layers from sensor layer to activity layer.

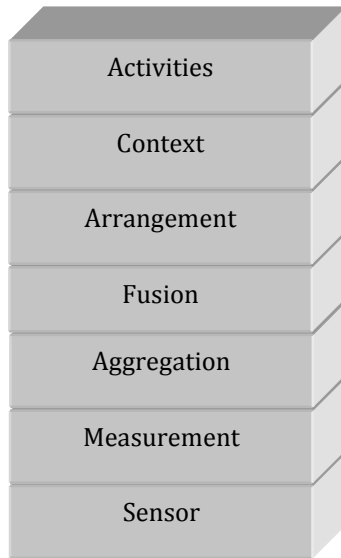


Fig. 1 Fundamental design of software development model for localization system

#### A. Description of Each Layer

In this section we describe each layer of the proposed software development model.

##### 1) Sensor Layer

Sensor layer contains hardware and software drivers for detecting various physical phenomena. The interface to this layer is usually very specific to the sensing technologies used. A sensor here could be represented by various sensor technologies such as WLAN card, RFID reader, inertial sensor, wireless sensor network tag, ultrasound sensor, and others.

##### 2) Measurement Layer

This layer provides data pre-processing capabilities to transform raw sensor data into a geometric measurement type along with uncertainty based on a model of the sensor that created the data, e.g. distance, angle, proximity, and fingerprint signature

##### 3) Aggregation Layer

Collects and caches data from sensors, deals with intermittently connected devices, provides communication/gateway to different technologies, e.g. wireless sensor network, and registration and authentication services, provides global time synchronisation service for clients.

##### 4) Fusion Layer

The Fusion layer provides localization data fusion processing of streams of measurement data. It implements algorithms to continually fuse streams of measurement data into a time-stamped probabilistic representation of positions and orientations of objects. We use a sequential non-linear Bayesian filter technique, implemented as particle filters [5], to estimate location in real multi-sensor environments. However, other implementations can also be used.

##### 5) Arrangement Layer

The layered model has also a mechanism to provide information about relationships between objects in the arrangement layer. Each object will be related to its corresponding environment description (map, floor plan). The layer has also responsibility to convert location information between absolute and relative coordinates which are used in the fusion layer

##### 6) Context Layer

This layer has responsibility for merging location information with other non-location, context information such as personal data (calendar, email, and contacts list), temperature, light level and others.

##### 7) Activity Layer

The layer provides an interface to specific application activities such as a system of rule-based triggers driving particular application scenarios.

#### B. Distributed Localization Model

As a requirement in the firefighter test case we implemented the localization model into a distributed system as shown in Fig 2. The system has the same architecture as our reference model and consists of four subsystems: main server, rich-client, thin-client, and provider. Each subsystem has specific characteristics which relate to its role in the system.

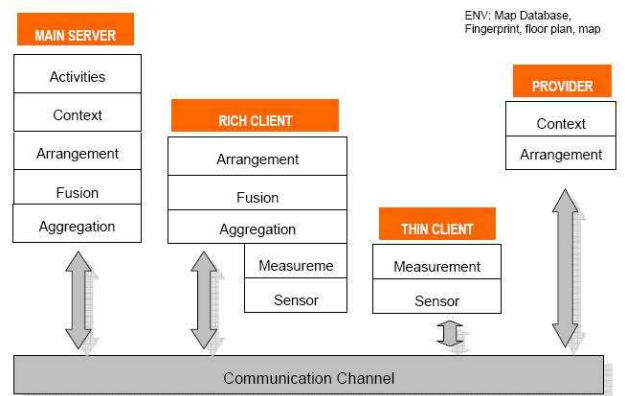


Fig. 2 Distributed Localization System

1. *Server* provides service to the client. It acts as the main computational machine. It implements the functionalities from aggregation layer up to activity layer.

2. *Rich Client* is a client which has processing capability. Basically, rich-clients have one or more sensors for gathering data of physical phenomena and capabilities to pre-process the data. Examples of this type of device are mobile PC and PDA.
3. *Thin Client* is a subsystem that mainly consists of sensors and very limited processing capability. Examples of this kind of device are Wifi tag, RFID tag.
4. *Provider* provides supporting information about environment description such as maps, floor plans, and fingerprints.

#### IV. APPLYING LOCALIZATION MODEL IN FIREFIGHTER SCENARIO

In this section we demonstrate in an example application as to how the distributed software development model we propose here can be used as a framework to develop distributed localization applications. The implementation is based in part on our work in the European Commission's FP6 funded project "WearIT@Work".

##### A. Scenario

The goal of the firefighter scenario is to guide firefighters entering a building when visibility is impaired due to smoke and also to guide them to find exits from the building. The system also aims at helping other firefighters to take the same path.

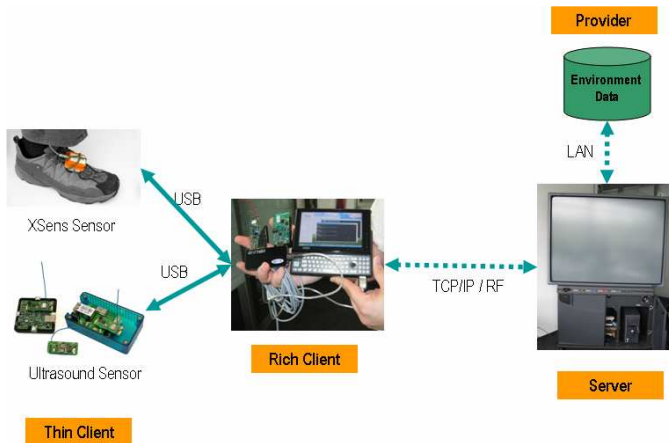


Fig. 3 System configuration for firefighter scenario

The configuration of the firefighter scenario used in the test case involves the firefighter command post outside the building (server), two types of sensors, provider, and one rich client as shown in Fig.3. The first type of sensor is the Relate Brick which uses ultrasound technology. Core processing and communication of the bricks is handled by a PIC18 microcontroller and an 868 MHz radio front-end [4]. The sensor board consists of four 40 kHz narrow-band ultrasound transducers, a temperature sensor and power supply using AAAA batteries. The transducers act both as receivers and transmitters. A USB port is used as a communications bridge

to connect the bricks to other devices like a portable or wearable computer within the network.

Firefighters deploy Relate Bricks along their path establishing an ad-hoc infrastructure for positioning and communication. Firefighters interact with this sensor network by way of wearable computing equipment and receive navigation information on a head-mounted display or over a headset.

Another sensor used is the MTX inertial measurement sensor (Xsens) which is a shoe-mounted device consisting of accelerometer, magnetometer (compass), and gyroscope sensors. It has an embedded processor capable of calculating roll, pitch and yaw in real time, as well outputting calibrated 3D linear acceleration, rate of turn (gyro) and earth magnetic field data. Measurements from the accelerometer sensor will be used to calculate displacement (step-length) of firefighters while magnetometer and gyroscope are used to determine the direction (heading) of the firefighter's movements.

The Provider supports the application by providing environment data related to the localization system. The typical environment data is a floor plan map. This map is needed by the fusion engine to relate between specific objects and their position on the floor plan map.

Fig. 4 shows the block diagram of the software architecture used in the experiment in relation to the layered architecture model of the localization framework. The two bottom most layers of the architecture gather measurement data of the physical phenomena i.e. acceleration from accelerometer sensors, heading (direction) for gyroscope sensors, and distance from ultrasound sensors.

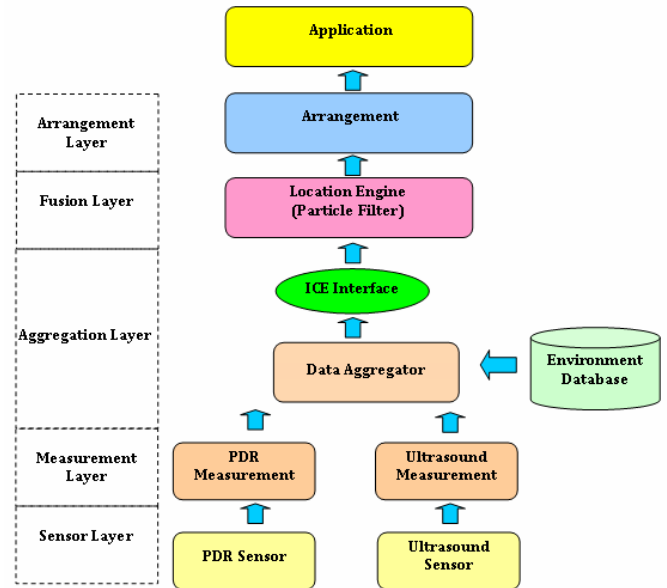


Fig. 4. Block Diagram of localization system and corresponding layer in the localization model

The Data Aggregator collects and caches both Pedestrian Dead Reckoning (PDR) data from the Xsens units and ultrasound data from the Relate Bricks. It also makes necessary modifications to the data format in order to make it

suitable for the location engine. As part of the aggregation layer, it has interfaces to communicate with the location engine.

The Location Engine uses a particle filter algorithm, a technique that implements recursive Bayesian Filtering using the sequential Monte Carlo method [5]. It is based on a set of random samples with weights, or particles, to represent the probability density. In addition, the location engine also uses a map filtering technique [5] and a back tracking technique [6] to improve accuracy of the tracked objects. Map filtering is implemented in a way to prevent particles, which represent a tracked object, to move through a wall or other solid object. Particles are only permitted to move along corridors or within rooms.

In the arrangement layer we associate object position resulted from the fusion engine (location engine) with the environment description used in the experiment. In a more advanced scenario this layer is also responsible for evaluating the relationship between two or more objects. In this scenario, we simply created a visualization which implements the context and activity layers, as we do not employ context and activity layers yet.

Determining the appropriate services, functionalities and interfaces in these layers are an important area of future research. These layers will provide standard interfaces through which most applications will access the location information.

This software architecture uses the distributed localization framework where computation of measurement data takes place in different sub-systems. A simplified diagram of the software architecture is shown in Fig 5.

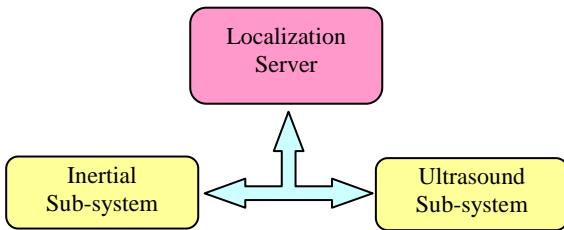


Fig. 5 Simplified block diagram of distributed localization framework implementation

Both subsystems have sensors as well as data preprocessing capabilities to process data and handle communication between subsystem and localization server.

### B. Result

A graphical visualization has been used in the test case in order to visualize both, object movement and its likelihood (particle). Fig. 6 shows visualization of the fire-fighter's movement in the building environment used in the test case. Fig. 6 has three different path colors representing the track of firefighters. The brown path is ground truth while the blue path is the estimated path as a result of the location engine using our particle filter and map filtering algorithms. The green line is visualization of firefighter's path after back the tracking algorithm is used in the data fusion process.



Fig. 6 Visualization of the fire-fighter scenario

With the experiment we have demonstrated and evaluated the performance of the software development model proposed in a real test case localization application. The software development model has been designed generically enough to support not only the test case application, but it can be considered as a common approach to develop new localization systems. The software model allows also aggregating multiple location sensor technologies, i.e. inertial sensors and ultrasound sensors, in this implementation in order to take advantage of each technology.

Finally, the model provides an opportunity for constant evolution of localization system development and deployment of new technologies into existing applications. The test case implementation has shown that the software model we proposed facilitates combining more than one location technology and a range of sensors. This will likely have impact on improving localization system's quality in the future.

## V. CONCLUSIONS

In this paper we have presented and demonstrated an implementation of our novel software development model for localization systems based on a distributed and hierarchical architecture. The model proposes a layered architecture, where each layer has interfaces to provide services to another layer. We have developed a test case application as an implementation of the framework to illustrate the advantages of this approach. The implementation of the test-case application has shown that this approach enables more flexible development of new localization applications which can have various different configurations. The model enables constant evolution of localization system development and deployment of new technologies into our current localization system.

Future research will be concentrated on analyzing and developing implementations of the two uppermost layers (context and activity) of the model. It will also be interesting to develop more advanced, three-dimensional visualization.

#### ACKNOWLEDGMENT

The authors wish to acknowledge the support of the Enterprise Ireland Proof of Concept through project PC/2007/0073 and the support of the EC FP6 IST WearIT@Work project (EC IST IP2003 0004216), [www.wearitatwork.com](http://www.wearitatwork.com), in funding the work reported in this paper.

#### REFERENCES

- [1] J. Hightower, et al. The Location Stack: A Layered Model for Location in Ubiquitous Computing, *The 4th IEEE Workshop on Mobile Computing Systems and Applications*, NY, USA, 2002
- [2] W. Lara, Universal Location Framework: A New Wireless Building Block, *Intel Developer Update Magazine*, USA, Feb 2003.
- [3] J. Hightower and G. Borriello, Location System for Ubiquitous Computing, *Computer*, 34(8) 57-66, August 2001.
- [4] M. Klann, T. Riedel, H. Gellersen, C. Fischer, M. Oppenheim, P. Lukowicz, G. Pirkel, K. Kunze, M. Beuster, M. Beigl, O. Visser, and M. Gerling, "Lifenet: an ad-hoc sensor network and wearable system to provide firefighters with navigation support," in *Adjunct Proceedings Ubicomp 2007*, Innsbruck, Austria, 2007.
- [5] Widyawan, M. Klepal, and D. Pesch, A bayesian approach for RF based indoor localization, in *Proceedings of The 4th IEEE International Symposium on Wireless Communication Systems 2007 (ISWCS 2007)*, Trondheim, Norway, October 16 2007
- [6] Widyawan, M. Klepal, and S. Beauregard. A backtracking particle filter for fusing building plans with PDR displacement data. In *Proceedings of the 5th Workshop on Positioning, Navigation and Communication (WPNC 2008)*, Hannover, Germany, March 27 2008.