A Lightweight Location System for Sentient Artefacts

Fahim Kawsar, Kaori Fujinami, Tatsuo Nakajima

Department of Information and Computer Science, Waseda University, Tokyo, Japan
{fahim,fujinami,tatsuo}@dcl.info.waseda.ac.jp

Abstract

This paper presents a location sensing system "Spreha" for sentient artefacts. Our approach towards context awareness is to use sensor augmented daily life objects for context sensing. Some of these artefacts are static in nature, like a cabinet, a bed, a refrigerator, etc. The basic idea in Spreha is to use these static artefacts as reference points for identifying the artefacts that are mobile in nature like a chair, a watch, a lamp, etc. This paper discusses about the architecture and current status of Spreha with illustrations on the performance.

1. Introduction

Location information has been considered as the most important context for contextual service provision. Though diverse researches are going on around the world on this field till date achievements are not significant for indoor location sensing. In outdoor environment Global Positioning System (GPS) has effectively been deployed widely. So, an obvious approach can be to extend the concept of GPS for indoor location sensing; that is using some static reference points for identifying mobile artefacts.

In our approach towards context aware environment, we focus on the environment itself. That means focusing on the building blocks of the environment and making them smart and context aware by capturing people's implicit interaction. We augment daily life artefacts like a chair, a table, a door, a mirror, a bed etc. with various kinds of sensors to capture contextual information. Our vision is to utilize these objects for value added services in addition to their primary roles. By augmenting sensors, we make these belongings (micro component of the environment) smart. Eventually this process recursively makes our environment smart and context aware in a bottom up approach. Based on our experiences of developing contextual applications integrating sentient artefacts we felt location information is very essential for sentient artefact based applications. However, one of the design principles of sentient artefact frameworks is not to use any dedicated sensing infrastructure [4]. Considering sentient artefacts' nature we have come up with the idea of using sentient artefacts for identifying sentient artefacts. From our observation we have seen there are various artefacts in our environment that are static in nature and we rarely move them, for example a refrigerator, a cooking oven, a room door/window etc. We exploited this static nature of these artefacts by using them as a reference point for identifying their peer mobile artefacts. We believe this approach is feasible, practical and economical in context aware environment as it eliminates the requirement of any dedicated sensing infrastructure. In this paper we discuss the design and implementation of this location sensing system.

The remaining paper is organized as follows: section 2 discusses about the design issues. In section 3 we have presented the architecture of Spreha. Section 4 discusses about the implementation and programming model of Spreha. Section 5 presents two applications that we have developed using Spreha. Section 6 depicts the performance of Spreha where as in section 7 we discuss on Spreha in general. In section 8 we have cited the related works, and finally section 9 concludes the paper.

2. Design Issues

In pervasive environment location information is very essential. This location information is exploited by the applications for various contextual services. However the location providers should exhibit some characteristics that are essential for pervasive environments. These are:

- **Transparency:** The location provider should gather location information in a transparent way without interference from the applications. Application will only be notified for location change event and must not be responsible for any network management.
- **Abstraction:** Heterogeneity is a common characteristic of pervasive environment. The location providers should cope with this heterogeneity issue of the underlying artefacts and should provide the location information in a unified generic way.
- **No Centralized Database**: The location information should not be stored in a centralized location; rather it should be distributed in all location providers.
- **Availability:** The location information should be available to the applications all the time regardless of the operating status of one or more location providers.
- **Privacy:** The location information should be protected from malicious client applications.

In addition to these issues, another important feature of location system that is specific to sentient artefact based computing is:

- **No Dedicated Infrastructure:** Location system should not be tightly coupled with any underlying sensing infrastructure. Instead we should be able to extract location information in an adhoc manner from sparsely distributed location providers.

We have used bluetooth as an underlying sensing technology in Spreha specifically for the following reasons:

- Wide acceptability and availability of Bluetooth in information appliances.
- Inexpensive in cost.
- Minimal Inference with other Radio Frequencies like IEEE 801.11a etc.
- As we only need to see the device and no bluetooth communication link need to establish, bluetooth's limitation of connecting to 7 slaves at a time has no effect.

These guidelines have been followed for designing Spreha. Also from our experience with application development we have observed that centimeter level accuracy is not needed for the development of contextual services, actually identifying proximity even with in few meters (2~4) is good enough for utilization. So, instead of accuracy essential design principles of Spreha are flexibility and simplicity.

3. Architecture

Spreha uses Bluetooth [12] as underlying technology for sensing the artefacts. A 48 Bit Bluetooth device address is used as the location identifier in Spreha, however a higher-level friendly name can also be used. The static location of the artefact is always a higher-level name such as Meeting Room, Fahim's Workspace etc. This static location is the identifier of the artefacts' location. For resolving the conflict when two or more hosts see the same artefact in their territory currently Time of Flight (TOF) is used, however Radio Signal Strength Indicator (RSSI) can also be activated where available.

In Spreha there is a predefined trust policy, which contains two attributes: public policy and private policy. Public policy means location information of the artefacts can be published publicly, whereas private policy means the opposite. Artefacts can provide their preferred policy during deployment time. As shown in figure 1 few logical components participate in Spreha; their roles are discussed in the following:

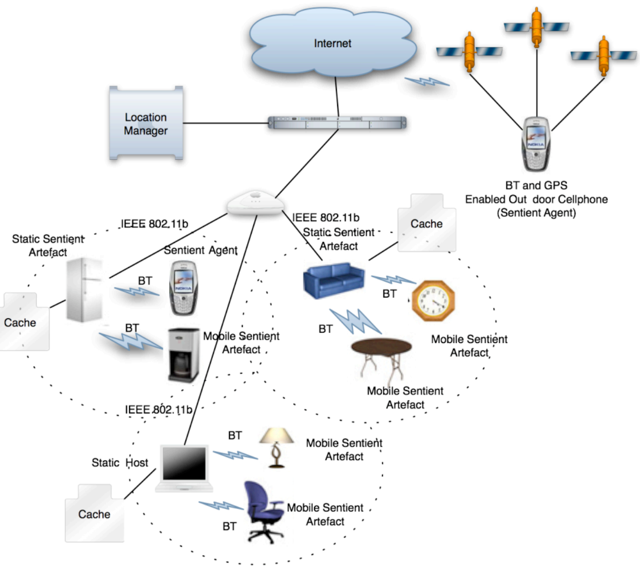

Figure 1: Architecture of Spreha

Location Manager: This is the central component that manages the location information of the artefacts. (Also each static host and static artefact manages location information locally) During deployment each artefacts register themselves to the manager. Each static artefact and static hosts periodically notify the manager about the artefacts information available to them. Application can query location manager for location information or can register for notification. On receiving new location information, it notifies the interested applications. Location manager resolves conflict when two or more hosts see the same artefact in their pico net by simply considering the minimum TOF and/or maximum RSSI for deducing artefacts location. However in case of out door sentient agents it communicates directly to receive the GPS information.

Static Sentient Artefact*:* This component acts as both a reference point and a location provider. Any sentient artefact that is considered to be stable in its location like a mirror, a cabinet, a couch, etc. can be considered as static artefact. These artefacts are augmented with bluetooth tag reader. It contributes to location sensing system by maintaining a cache of nearby sentient peers that is periodically updated by running the discovery service embedded in it. The discovery service discovers the near by peers within its pico net. The cache also contains the RSSI and/or TOF. Whenever the cache state is changed it notifies the location manager and to the applications subscribed to them. During the deployment of the artefact it specifies its name, role as a location provider, its static location and its security policy to resource manager.

Mobile Sentient Artefact*:* This component is the ordinary sentient artefact that is mobile in nature like a chair, a watch, etc. A bluetooth tag is embedded in it. Static artefacts and/or the static hosts identify these tags and notify the location manager. During deployment these artefacts specifies their mobile role, name and security policy.

Static Host: This component is an ordinary location provider embedded with a bluetooth tag reader and runs the discovery service periodically and maintains a cache of seen artefacts. Whenever cache state is changed, location manager and the subscribed client applications are notified. During deployment it specifies its role as static host and its static location.

Sentient Agent*:* This is a special component that assumes to be run in a handheld device owned by a person. Spreha assumes that a person will carry this device. During initial deployment the agent should register its name, IP address, and security policy. Hosts identify this agent when it is in their designated pico net and notifies location manager. However if the agent's location information is missing when queried by applications, then location manager communicates directly with the agent running in the handheld, and agent uses the GPS to retrieve its location information and notifies the manager. Thus locating nomadic people is supported in Spreha.

4. Implementation and Programming Model

Spreha is basically a part of the middleware for sentient artefact based computing titled "Prottoy" [6], which is composed of three core components. The core components are:

1. Resource Manager: Responsible for resource discovery, managing location information and reconfiguration of the underlying environment.
2. Artefact Wrapper: Responsible for encapsulating artefacts and offering artefact service and context information to applications.
3. Virtual Artefact: Responsible for providing unified interface to applications for interacting with the underlying layers.

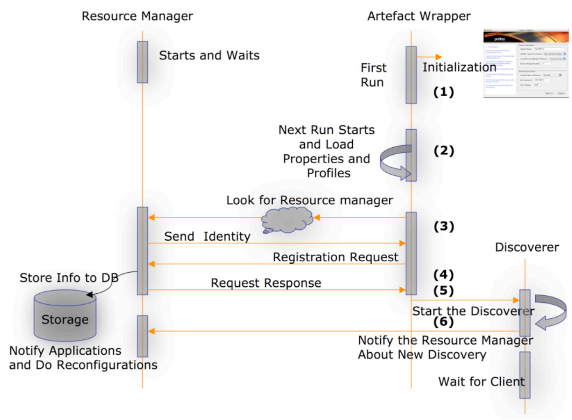

Figure 2: Artefact Deployment in Spreha

The location manager component of Spreha is a module of the Resource Manager of Prottoy that notifies location information to clients and responses the application query. It also dynamically reconfigures the underlying environment topology based on location information. The artefacts (static artefact, static host, mobile artefacts and sentient agents) are deployed in the environment using the Artefact Wrapper component of

Prottoy that encapsulates artefacts functionality and represents the physical artefact in the application space. The artefacts deployment process is depicted in figure 2 where in (1)-(4) the artefacts communicates with the resource manager to provide their property information, location and security policy as specified in previous section. The (5) – (6) the artefacts start the discovery process and notify the manager periodically. The artefacts are deployed with artefact wrapper GUI component of Prottoy. For detail of this process please check the reference [6].

For application developers Spreha provides few API as shown in the following table.

Table 1: API for Interacting with Spreha

`1. Artefact[] getArtefactByProfile( String profile)`
`2. Artefact[]getArtefactByLocation( String location)`
`3. void addLocationListener( Object source, String callback)`
`4. void removeLocationListener()`

The application developer uses the first two APIs for querying *location manager only* for artefact availability based on profile and location. Here profile is the role that an artefact plays. Examples of profiles are: light-service, display-service, environment-attribute, state of use, etc. An artefact can implement one or more profiles.

As it is shown in figure 2, the location manager periodically receives the location information from static artefacts, so when application asks for an artefact location using these two API it checks its local cache and notifies the application. The following code segment demonstrates these artefacts use. Here `QueryProcessor` is the Spreha interface provided to application developer. These two API returns an array of artefacts that matches the location or profile.

```
Artefact [] artefact = new
QueryProcessor().getArtefactByLocation("L-50");
if(artefact!=null){
  //extract artefact info }

Artefact [] artefact = new
QueryProcessor().getArtefactByProfile("Light");
if(artefact !=null){
  //extract artefact info }
```

The next two APIs are for subscribing to the location manager, static artefacts and static hosts for artefacts location information. These APIs are used using the virtual artefact component of Prottoy. The callback receives an array of artefacts with their location. Here developers provide the callback function name by themselves. We have used Reflection techniques for this purpose thus eliminated any inherent dependency of application on the framework. The following code segment shows this API usage

```
VirtualArtefact mirror = new
            VirtualArtefact("display-Service");
if(mirror.status){
 mirror.addLocationListener(this,"callback");
}
mirror.removeLocationListener();
QueryProcessor.addLocationListener
                       (artefact.getName());
```

Here we have used `VirtualArtefact` instance to subscribe to the mirror artefact for receiving its cache of location information and next we have subscribed to location manager for a specific artefact's location information. Here `Artefact.getName()` is used as parameter which can be retrieved by `getArtefactByLocation()`.The detail of the data structure of Spreha is mention in [6].

Another important issue is that Spreha does not implement any location model on top its sensing system. Any suitable location model can be used on top of Spreha to represent the physical world, for instance: we have mentioned in section 6 that it is necessary to do a prior layout design before deploying Spreha, if each static host and static artefacts static location is organized in a predefined hierarchical manner we can easily represent any virtual model of the physical world. For instance a visualization of the model proposed by Satoh [8] that consider containment relationship is shown in the following figure 3.

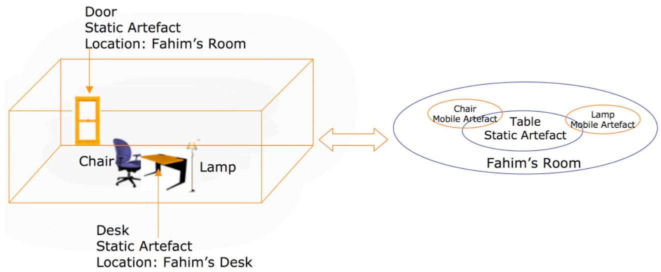

Figure 3: Illustration of a Conceptual Model

Here we have arranged the static hosts/artefacts hierarchically; when we received location information of an artefact we can easily deduce the location of the artefact in the physical world considering containment relationship. For example if we receive the lamp's location is Fahim's desk from the desk then we can easily deduce that the lamp is in Fahim's room as the desk is in Fahim's Room.

5. Sample Application

In this section we will present two applications that we have deployed using Spreha's location sensing capability. The applications are built on top of Prottoy [6].

5.1. SoLite

This is a very simple application that employs only two mobile artefacts namely a stand light, a chair and a static artefact: a desk. If the chair and the stand lights are in the desks location, the light is automatically turned on/off based on the ambient light sensitivity of the surrounding and the presence of the user sensed by the state of use of the chair (sitting/not sitting).

5.1.1. Functionality Mapping

The scenario required following functional mapping:

Table 2: Functionality Mapping for SoLite

	Scenario Functionality	Required Capability	Augmented Artefact Used
D E S K S I D E	Changing workspace environment	Capturing neighborhood brightness	Desk lamp with photo sensor
		Capturing users presence	Chair augmented with sensor
		Location of the chair and lamp	Desk augmented with bluetooth tag reader

5.1.2. *Component View*

SoLite uses the following component:

1. **Sentient Lamp**: It is a traditional stand lamp that is augmented an ambient light sensor. The lamp is connected to the power line using X10 module. The light sensor is used to track the environment's light level.
2. **Sentient Chair:** An ordinary chair augmented with 4 force sensor on the seat and two photo sensor on the back to identify the state of use of the chair: sitting or not sitting.
3. **Sentient Desk:** An ordinary desk augmented with a bluetooth tag reader and act as a static sentient artefact.

5.1.3. *Functional View*

Figure 4 shows the SoLite in operation. Its control flow is also very simple.

1. The application activates when both the light and chair are in the desk location
2. When the environment gets darker, the application initiates to sense the chair state.
3. If chair is in use it turns on the lamp with appropriate dimming based on the sensed information.
4. If the chair is not available or not in the location of the lamp then application ignores the actuation.
5. If another chair/tool is available instead of original chair (when it is moved) that serves the same context information, then location manger dynamically change the application binding to new chair/tool from migrated chair.

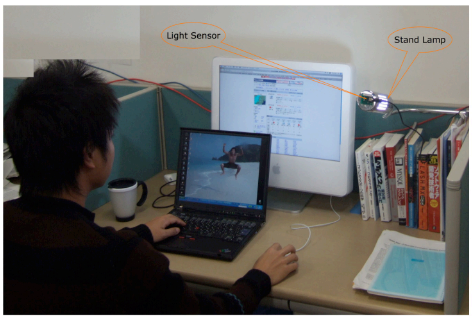

Figure 4: SoLite in Operation

5.1.4. *Observation*

The basic goal of this simple application was to demonstrate the applicability of Spreha. Since a desk is rarely moved we can consider it as a static sentient artefact where as the chair and the lamp are being considered as mobile artefacts due to their inherent mobility nature. We have found that the application run very smoothly using Spreha for identifying the location of the underlying artefacts and actuating the actions accordingly. Spreha's approach was successful in this application.

5.2. Auto Presenter

This application is designed for assisting conference attendees in the poster sessions at conferences. Though the application is very simple, it is quite useful for conferences. The basic idea is to provide the attendees with a handheld device, which can run a small video clip about the poster content that the attendee is proximity wise nearby. As we found very often in the conferences that the poster presenter is not available during the technical sessions, such application can help the attendees to have a visual explanation of some of the posters they are interested in while presenter is absent. This tool can make the poster sessions more attractive.

5.2.1. *Functionality Mapping*

The application requires the following functional mapping.

Table 3: Functionality Mapping for Auto Presenter

	Scenario Functionality	**Required Capability**	**Augmented Artefact Used**
R O O M	Play a video clip on clients PDA.	Detecting location of clients PDA.	PDA augments with tag and Poster Panels augmented with Bluetooth tag reader.

5.2.2. *Component View*

The following components are used in the Auto Presenter.

1. **Poster Panels:** Each poster panel is augmented with a bluetooth tag reader. In our case we have used Bluetooth USB Dongle associated with each poster panel. Each poster acts as a static host.

2. **PDA:** The clients PDA is an ordinary PDA that acts as s mobile sentient artefact.

5.2.3. *Functional View:*

Figure 5 shows the Auto Presenter in operation. The control flow of the application is as follows:

1. The Auto Presenter application runs remotely.
2. The application subscribes to the location manager for location information of the PDA.
3. Each poster panel has a static location. When the panel identifies the PDA it notifies the location manager, which in turns notifies the application.
4. On receiving the PDA's location the application executes the display service of the PDA s that have changed their location and are still in the poster room. Each static location name is the name of the poster itself and for each poster a video clip is put in the PDA. So on receiving a specific poster name the PDA plays the associated video clip.

Figure 5: Auto Presenter in Operation

5.2.4. *Observation*

We have found the performance of the application is not satisfactory. The location sensing system was very unstable. Especially because of the congested posters it was very hard to distinguish the posters by analyzing the time of flight. However except the location sensing conflict, the application performs pretty well from overall functionality point of view. Due to location sensing conflict the PDA swaps movie clips undesirably but this conforms that the functionality of the system was maintained (to show selective video clip based on location).

6. Performance Analysis

In a real environment we have found that bluetooth performance was not always satisfactory. Especially if the static hosts and static sentient artefacts are located in congested manner than it is very difficult to infer the actual location of the mobile artefacts. The following figure 6 shows the relation of the distance and time of flight received from a mobile artefact by a static host.

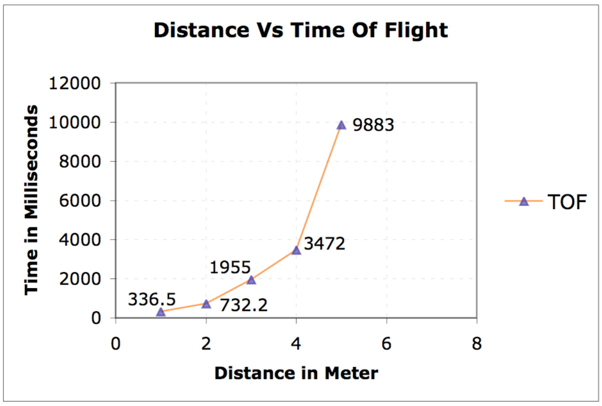

Figure 6: Bluetooth Performance in Spreha

As it is shown, that beyond 8~9 meters (although according to bluetooth specification 10 meter is the operational range [12]), we hardly receive any response from the mobile artefacts during discovery, also the time of flight (TOF) measure is fluctuating. The data shown here is the average best case when tested by 100 runs with mobile artefacts sparsely distributed at varying distances. Also if two static hosts/artefacts are located nearby it is difficult to select the proper one by only calculating the TOF, because sometimes it leads to wrong prediction. In figure 7 we have demonstrated 3 cases for clarification of this issue, Two static artefacts are located by a difference of 8~12 meters and one mobile artefact is at varying distance from these two artefacts. We have run the discovery service in both the static artefacts 100 times simultaneously and found the result depicted in the pie chart for each case. We have observed that if the static artefacts are closely located then the number of wrong prediction increases and vice versa.

As we have specified in the Auto Presenter application, that due to this fluctuating nature of TOF, the applications functionality could not be achieved completely. For improving the performance, one alternative is to combine RSSI with TOF for better prediction. However, in the current implementation we have not considered RSSI.

It is necessary that the static hosts and static artefacts are arranged in a disperse manner for proper location sensing. In general, this is not a shortcoming of the location sensing system because we can distribute the artefacts in a way that they do not conflict with each other and it is logical, for instance consider a kitchen we can have several sentient artefacts that are static in nature like a cooker, a refrigerator, a cabinet, etc. that are closely arranged. However, we can use only one of them as a reference point in the kitchen for location discovery, this approach is practical and economical. So the only constrain to use Spreha is that we need a prior design and layout of the environment for deciding the artefacts that can play the role of static host or static sentient artefact.

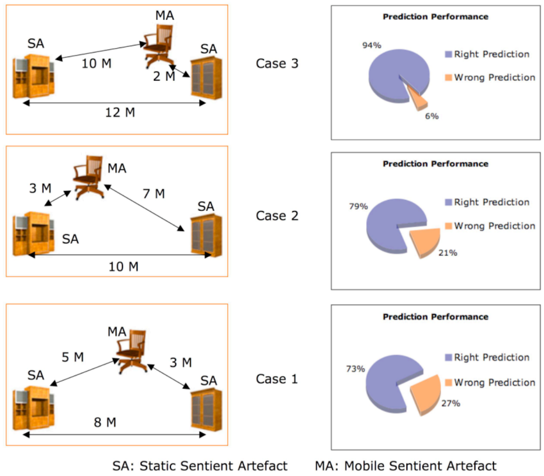

Figure 7: Relation between TOF and Prediction

7. Discussion

From design principle point of view Spreha satisfies the transparency and abstraction requirement by using the artefact wrapper component of Prottoy [6]. In Spreha location information is stored centrally in location manager and each static host and static sentient artefact also host their own location information. Applications can query location manager for retrieving artefact location where as can subscribe to location manager, static host and static artefacts for location information. From this point of view Spreha's approach is a hybrid one between centralized and distributed data storage. Because of this hybrid organization in Spreha the location information is always available either from location manager or from static location providers. In Spreha we have used a simple trust model that ensures the protection of artefact information. If an artefact location policy is private, that artefact location information is not dispatched to application by any of the location providers or location manager. Although Spreha does not provide any mechanism to ensure the dispatched location information is not misused in application space.

Spreha does not exploit any dedicated sensing infrastructure for location sensing. We use sentient artefacts with augmented services for location sensing in an adhoc manner. From this point of view: readers may be confused about what we mean by dedicated. Our proposition here is that each artefact has its primary role in our everyday life. We are keeping that role intact while using it for location sensing. So the underlying infrastructure is the sentient artefacts not any sensor nodes deployed only for location sensing like active bat, ubiSense or cricket [10,13,7]. Thus the major advantages of Spreha can be listed as follows:

1. The notion of sentient artefact as location node is the primary advantage of Spreha.
2. Cost of Location Node is minimal as no external sensing system is necessary. The value added services of sentient artefact nullifies the location system cost as the location system components (bluetooth tag and reader) are parts of the sentient artefacts. If the artefacts are not bluetooth enabled, then the cost of the system is the summation of the

Bluetooth configuration cost for each mobile and static artefact.

The applications mentioned in section 5 illustrate these advantages. We have demonstrated how the sentient artefacts are performing the multiple sentient roles beyond its primary role. By doing so we have eliminated the necessity of any external location system. If we deploy these applications with other location system, we need to use an external sensing system on top/bottom of the sentient artefacts, however in Spreha the sentient artefact notion itself provides this facility.

It is understandable that Spreha is a very thin lightweight location-sensing system suitable for sentient artefacts based applications. By lightweight we emphasize that, Spreha does not need any external sensing infrastructure except the sentient artefact itself for location sensing. It is impractical to think of Spreha as a general context aware location system. Although the underlying bluetooth technology depicted fluctuating nature, we believe our proposition of using static artefacts, as reference point is logical and economical. Bluetooth can be replaced with other technology like IEEE 802.11 wireless standards. For example Ekahau tags and tag readers (http://www.ekakhau.com) can be used in Spreha without any major modification. The strength of Spreha is not the sensing technology but the idea of using sentient artefacts as a location reference point. So the issues of other bluetooth or proximity sensing system should not be compared here. Since the location information is the static location of static artefacts, the drawbacks of traditional proximity sensing techniques like lateration should also not be considered here, because in Spreha the location is dynamically configured when the static artefact is deployed in the environment. Because of these the reconfiguration role of location manager partially depends on static artefacts. So if a static artefact changes its location, it is needed to restart the static artefact with new location.

We believe Spreha's proposition poses interesting issues for further investigation. We consider dedicated infrastructures are not applicable in a domestic environment. Our claim is further justified by the recent proliferation of Place Lab approach [9] of using existing networks for location detection. Our idea is actually inherited from their concept of using WiFi/RF base stations as reference point for location detection. We extend their concept by using sentient artefact notion for value added services. But the artefact actually is an RF base station. So from this point of view Spreha can be thought of as an extension emerged from Place Lab. Another important issue is that Spreha does not implement any location model on top its sensing system. Any suitable location model can be used on top of Spreha to represent the physical world.

8. Related Work

Considering Spreha's proposition, comparing Spreha with other location sensing system may seem ambiguous. The reason is Spreha introduces sentient artefact in location provider dimension but it is using bluetooth as underlying sensing technique. And using bluetooth for indoor location sensing is not a new observation as it has already been explored in [1,5]. So from this point of view we cannot actually compare Spreha with other indoor sensing. Spreha's contribution is in introducing the novel notion of sentient artefact as reference point. On the other hand Spreha does not implement any location model as proposed in numerous literature [2,3,8,10,11]. The distinction of Spreha with other indoor location system is an intellectual one because of the utilization of sentient artefact instead of dedicated infrastructure. For example there are numerous indoor location system that make use of ultrasonic [7,10], infrared [11], ultra-wideband radio [13]. All these systems require a hardware infrastructure be installed in the environment. Most importantly these systems are generally expensive, costing thousands to tens of thousands of US dollars for a 1000 square meter installation. These systems primarily focus on optimizing accuracy rather than wide-scale deployment. We consider these systems are not suitable for sentient artefact based computing because of such inherent dependency on infrastructure. Place Lab proposes using RF/WiFi base stations as reference points. [9] Basically we can think, Spreha augment their idea by embedding the base station in the sentient artefacts that are static in nature. Though using RF access points as reference is inherited from Place Lab, Spreha introduces few features that are missing in Place Lab, like artefact end security policy, distribution of location information in static hosts and artefacts, role of location manager and the notion of sentient agent for seamless change between bluetooth to GPS usage.

9. Conclusions

In this paper we have proposed a lightweight location sensing system suitable for sentient artefact based computing. The inherent domestic domain nature of sentient artefact computing requires the elimination of dedicated infrastructure for location sensing, we argue Spreha is good answer towards this issue. It is logical and economical to adopt the approach of using static artefacts as a reference point for sensing the peer artefacts location. Although due to bluetooth fluctuating performance, the real time performance of Spreha is not as expected, but still it is acceptable for most of the applications. We hope in near future we will incorporate RSSI with TOF for better prediction of locations of mobile artefacts to improve the performance of Spreha further.

References

[1] Anastasi, G., Bandelloni, R., Conti, M., Delmastro. F., Gregori. E., and Mainetto, G. "Experimenting an Indoor Bluetooth-Based Positioning Service" In Proceedings pf the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03), 2003.

[2] Beigl, M., Zimmer, T., and Decker, C."A location model for communicating and processing of context." Personal and Ubiquitous Computing, 6(5–6): 341–357, 2002.

[3] Burnett, M., Prekop, P. and Rainsford. C. "Intimate location modeling for context aware computing." In Proceedings of the Workshop on Location Modeling for Ubiquitous Computing, in UbiComp2001, pages 77–82, 2001.

[4] Fujinami, K. and Nakajima, T. "Augmentation of Everyday Artefact for Context-Aware Applications' Building Blocks", In Proceedings of the Workshop on Smart Object Systems In Conjunction with the Seventh International Conference on Ubiquitous Computing (UbiComp 2005) 2005.

[5] Hallberg, J., Nilsson, M., and Synnes, K. "Positioning with Bluetooth." In Proceedings of the 10th International Conference on Telecommunications (ICT 2003). 2003.

[6] Kawsar, F., Fujinami, K. and Nakajima, T. "A Middleware for Sentient Environments", In Proceedings of The 2005 IFIP International Conference on Embedded And Ubiquitous Computing (EUC-05), 2005.

[7] Priyantha, N. B., Chakaraborty, A. and Balakrishnan, H. "The Cricket Location-Support System" In Proceedings of MOBICOM 2000.

[8] Satoh, I. "A Location Model for Pervasive Computing Environment", In Proceedings of the Third Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2005). 2005

[9] Schilit, B., LaMarca, A., Borrirllo, G., Griswold, W., Mcdonald, D., Lazowska, E., Balachandran , A., Hong, J. & Iversion. "V. Challenge: Ubiquitous Location-Aware Computing and the Place Lab Initiative" In Proceedings of the First ACM International Workshop on Wireless Mobile Applications and Services on WLAN (WMASH), 2003

[10] Want, R., Hopper, A., Falcao, V. and Gibbons, J. "The Active Badge Location System", ACM Transactions on Information Systems, 10, 91-102. 1992

[11] Ward, A., Jones, A. and Hopper, A. "New Location Technique for the Active Office", IEEE Personal Communications, 4, 42-47. 1997

[12] Specification of the Bluetooth Core System. www.bluetooth.org

[13] www.ubisense.net