Full Scale Software Support on Mobile Lightweight Devices by Utilization of All Types of Wireless Technologies

Ondrej Krejcar

VSB Technical University of Ostrava, Center for Applied Cybernetics, Department of measurement and control, 17. Listopadu 15, 70833 Ostrava Poruba, Czech Republic
Ondrej.Krejcar@remoteworld.net

Abstract. New kind of mobile lightweight devices can run full scale applications with same comfort as on desktop devices only with several limitations. One of them is insufficient transfer speed on wireless connectivity. Main area of interest is in a model of a radio-frequency based system enhancement for locating and tracking users of a mobile information system. The experimental framework prototype uses a wireless network infrastructure to let a mobile lightweight device determine its indoor or outdoor position. User location is used for data prebuffering and pushing information from server to user's PDA. All server data is saved as artifacts along with its position information in building or larger area environment. The accessing of prebuffered data on mobile lightweight device can highly improve response time needed to view large multimedia data. This fact can help with design of new full scale applications for mobile lightweight devices.

Keywords: Mobile Lightweight Device; Localization; Prebuffering; Response Time; Area Definition.

1 Introduction

The usage of various mobile wireless technologies and mobile embedded devices has been increasing dramatically every year and would be growing in the following years. This will lead to the rise of new application domains in network-connected PDAs (Personal Digital Assistants) that provide more or less the same functionality as their desktop application equivalents. The idea of full scale applications pursuable on mobile lightweight devices is based on current hi-tech devices with large scale display, large memory capabilities, and wide spectrum of network standards plus embedded GPS module. Example of such devices is HTC Touch HD.

Users of these portable devices use them all time in context of their life (e.g. moving, searching, alerting, scheduling, writing, etc.). Context is relevant to the mobile user, because in a mobile environment the context is often very dynamic and the user interacts differently with the applications on his mobile device when the context is different [1].

My recent research of context-aware computing has been restricted to location-aware computing for mobile applications using a WiFi network (LBS Location Based Services). The information about basic concept and technologies of user localization such as LBS, Searching for WiFi AP) can be found in my article [2]. On localization basis, I created a special framework called PDPT (Predictive Data Push Technology) which can improve a usage of large data artifacts of mobile devices [3]. I used continual user position information to determine a predictive user position. The data artifacts linked to user predicted position are prebuffered to user mobile device. When user arrives to position which was correctly determined by PDPT Core, the data artifacts are in local memory of PDA. The time to display the artifacts from local memory is much shorter than in case of remotely requested artifact.

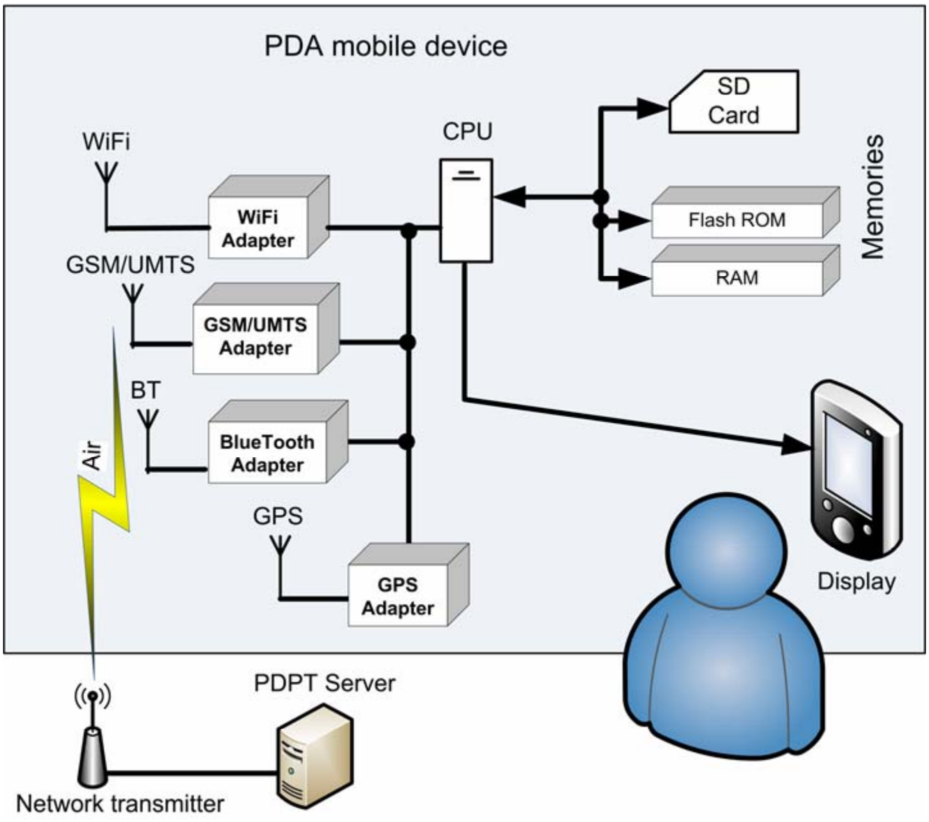


Fig. 1. Wireless networks and GPS sensor localization possibilities on mobile devices

The idea of prebuffering may not be only one application method for user position knowledge. As well as WiFi is not only one wireless network to use for localization of user device. WiFi has advantage in speed in indoor positioning therefore the GSM/UMTS can be used in outdoor [Fig. 1]. The GPS sensor is also embedded in several types of current mobile devices, or it can be plugged by SDIO or BT interface.

I would like to describe a position obtaining from wireless networks background in the beginning of next chapter to give a reader more information about these themes.

2 The PDPT Framework and PDPT Core

The general principle of my simple localization states that if a WiFi-enabled mobile device is close to such a stationary device – Access Point (AP) it may “ask” the provider’s location position by setting up a WiFi connection. If position of the AP is known, the position of mobile device is within a range of this location provider. This range depends on type of WiFi AP. The Cisco APs are used in my test environment at Campus of Technical University of Ostrava. I performed measurements on these APs to get signal strength (SS) characteristics and a combination of them called “super ideal characteristic”. More details can be found in chapter 2.3 [5]. The computed equation for Super-Ideal characteristic is taken as basic equation for PDPT Core to compute the real distance from WiFi SS.

From this super ideal characteristic it is also evident the signal strength is present only to 30 meters of distance from base station. This small range is caused by using of Cisco APs. These APs has only 2 dB WiFi omnidirectional antenna. Granularity of location can be improved by triangulation of two or more visible WiFi APs. The PDA client will support the application in automatically retrieving location information from nearby location providers, and in interacting with the server. Naturally, this principle can be applied to other wireless technologies like Bluetooth, GSM or WiMAX.

To let a mobile device determine its own position is needed to have a WiFi adapter still powered on. This fact provides a small limitation of use of mobile devices. The complex test with several types of battery is described in my article [4] in chapter (3). The test results with a possibly to use a PDA with turned on WiFi adapter for a period of about 5 hours.

2.1 The Need of Predictive Data Push Technology

PDPT framework is based on a model of location-aware enhancement, which I have used in created system. This technique is useful in framework to increase the real dataflow from wireless access point (server side) to PDA (client side). Primary dataflow is enlarged by data prebuffering. PDPT pushes the data from SQL database to clients PDA to be helpful when user comes at final location which was expected by PDPT Core. The benefit of PDPT consists in time delay reducing needed to display desired artifacts requested by a user from PDA. This delay may vary from a few seconds to number of minutes. Theoretical background and tests were needed to determine an average artifact size for which the PDPT technique is useful. First of all the maximum response time of an application (PDPT Client) for user was needed to be specified.

Nielsen [6] specified the maximum response time for an application to 10 seconds [7]. During this time the user was focused on the application and was willing to wait for an answer. The book is over 20 years old (published in 1994). I suppose the modern user of mobile devices is more impatient so the stated value of 10 second will be shorter. This is for me even better, because my framework is more usable. I used this time period (10 second) to calculate the maximum possible data size of a file transferred from server to client (during this period). If transfers speed wary from 80 to 160 kB/s the result file size wary from 800 to 1600 kB. More details about the facts of slow transfer speed on mobile devices can be found in chapter 2.5 [5].

The next step was an average artifact size definition. I use a network architecture building plan as sample database, which contained 100 files of average size of 470 kB. The client application can download during the 10 second period from 2 to 3 artifacts. The problem is the long time delay in displaying of artifacts in some original file types. It is needed to use only basic data formats, which can be displayed by PDA natively (bmp, jpg, wav, mpg, etc.) without any additional striking time consumption.

The final result of our real tests and consequential calculations is definition of artifact size to average value of 500 kB. The buffer size may differ from 50 to 100 MB in case of 100 to 200 artifacts.

2.2 From Data Collection to Localization

A first key step of the PDPT is a data collection phase. I record information about the radio signals as a function of a user's location. The signal information is used to construct and validate models for signal propagation. Among other information, the WaveLAN NIC makes the signal strength (SS) available. SS is reported to units of dBm. Each time the broadcast packet is received the WaveLAN driver extracts the SS information from the WaveLAN firmware. Then it makes the information available to user-level applications via system calls.

If the mobile device knows the position of the stationary device (transmitter), it also knows that its own position is within a range of this location provider. The typical range wary from 30 to 100 m in WiFi case, respectively 50 m in BT case or 30 km

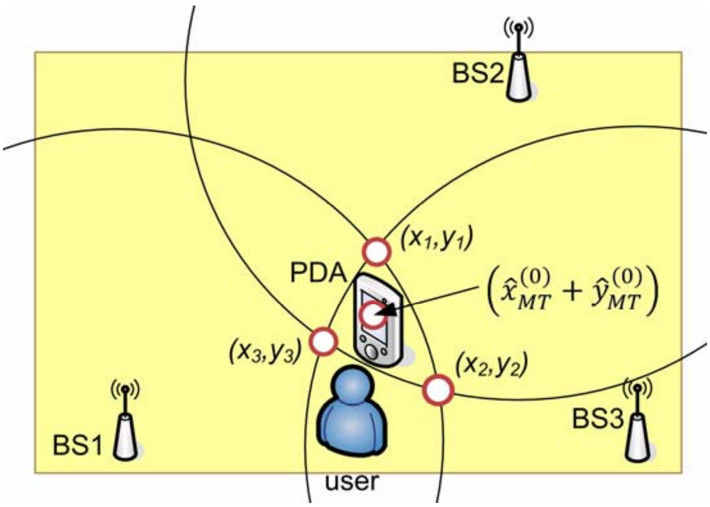


Fig. 2. Localization principle – triangulation

for GSM. Granularity of location can be improved by triangulation of two or more visible APs (Access Points). The PDA client currently supports the application in automatically retrieving location information from nearby WiFi location providers, and in interacting with the PDPT server. Naturally, this principle can be applied to other wireless technologies like BT, GSM, UMTS or WiMAX. The application (locator) is implemented in C# language using the MS Visual Studio .NET with .NET Compact Framework and a special OpenNETCF library enhancement. Schema on figure [Fig. 2] describes a localization process. The mobile client gets the SS info of three BSs (Base Stations) with some inaccuracy. Circles around the BSs are crossed in red points on figure. The intersection red point (centre of three) is the best computed location of mobile user. The user track is also computed from these measured SS intensity levels and stored in database for later use by PDPT Core. This idea is applicable in case of WiFi as well as BT and GSM networks.

In previous research, I focused only to use of WiFi networks while the other wireless possibilities remained without a proper notice. Now I made an enhancement of Locator component of PDPT framework [Fig. 5] to allow operate with BT and GSM networks.

In BT network case, the position of BT APs must be known to allow the position determination. To collect BT APs position info in outdoor environment, the GPS can be used. For indoor area, the GIS (Geographic Information System) software with buildings map must be used to measure exact position of BT AP against to local environment. To manage with BT hardware of mobile device another library InTheHand 32Feet.NET is used. The source code has a simple implementation:

Example of a Locator Source Code – Scanning the nearby for BT APs.

```
using InTheHand.Net.Bluetooth;

BluetoothClient bc = new BluetoothClient();
BluetoothDeviceInfo[] bdi = bc.DiscoverDevices();

foreach (BluetoothDeviceInfo BTdi in bdi)
{
  drDataRow = dtVisibleAP.NewRow();
  drDataRow["AP_name"] = BTdi.DeviceName.ToString();
  drDataRow["MAC_AP"] = BTdi.DeviceAddress.ToString();
  drDataRow["Signal_Strength"] = BTDi.Rssi;
  drDataRow["Date_Time"] = DateTime.Now;
  drDataRow["AP_type"] = AP_type.Bluetooth;
  dtVisibleAP.Rows.Add(drDataRow);
}
```

GSM network is not local network but a cellular network. The problem is in position information of GSM BTSs (Base Transceiver Stations). The operator doesn't provide any such information. One of possible solutions is based on unofficial BTSs lists which can be found on internet. The lists are typically available in HTML, TXT or CSV formats. The medium rate for BTs with GPS position information is about

90 % of all BTs in European countries. In case of PDPT Framework, the list must be converted to PDPT server database – GSM_BTS table [Fig. 3].

In all three described cases of nearby BSs scanning, the data are saved to Locator Table in PDPT server DB [Fig. 3]. Data are processed from Locator Table throw the PDPT Core to Position Table. The processing techniques depend on selected wireless network. WiFi and BT network provide all visible APs nearby the user. From list of these APs is computed actual position (by triangulation [Fig. 2]).

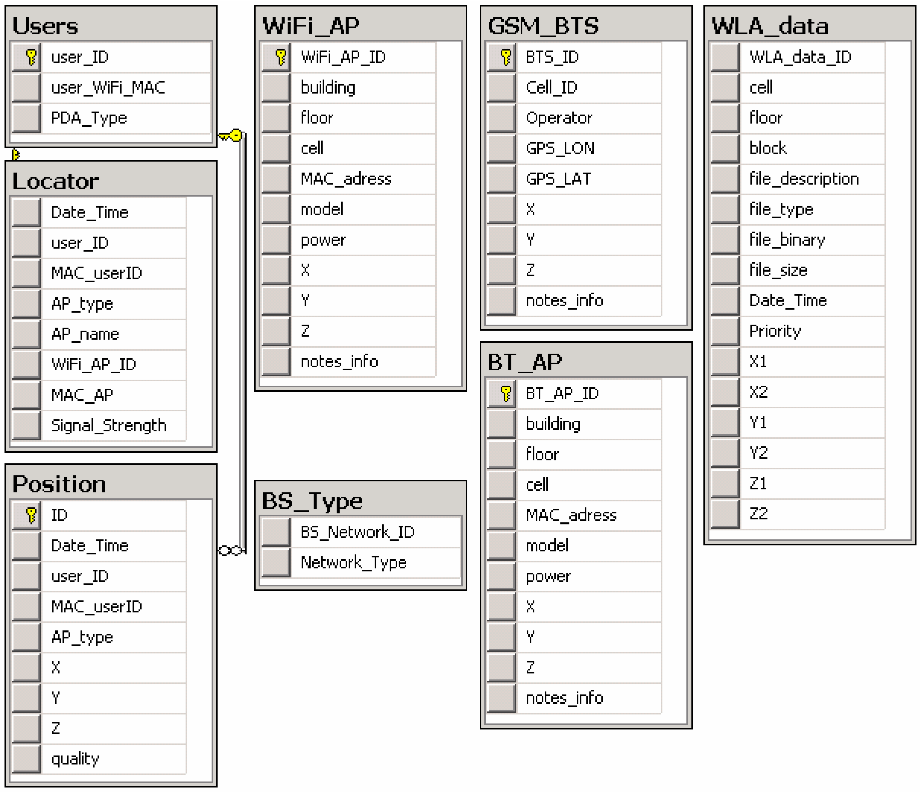


Fig. 3. PDPT server DB (Data Base) - New database architecture

Mobile devices with windows mobile operation system do not provide any GSM info to .NET Compact Framework. Even any special framework as in previous two cases is not known for me until now. Only possibility is in use of RIL (Radio Interface Layer) library. This library is divided into two separate components, a RIL Driver and a RIL Proxy. The RIL Driver processes radio commands and events. The RIL Proxy performs arbitration between multiple clients for access to the single RIL driver. When a module calls the RIL to get the signal strength, the function call immediately returns a response identifier. The RIL uses the function response callback to convey signal strength information to the module [Fig. 4].

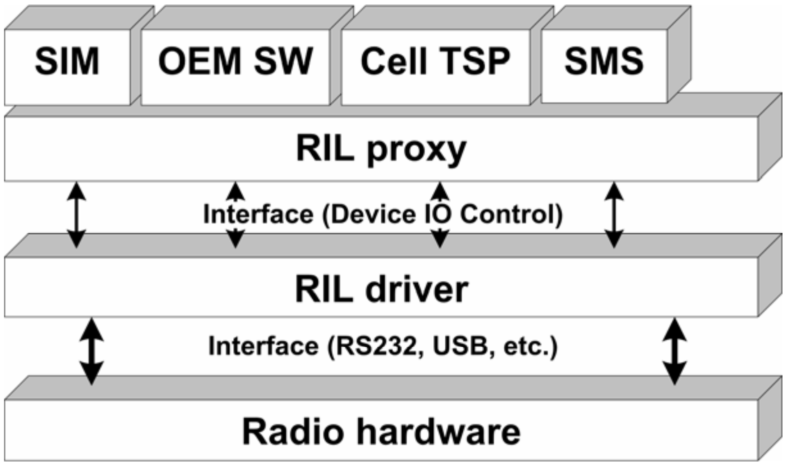


Fig. 4. Radio Interface Layer Architecture

Example of a Locator Source Code – retrieving the GSM BTSs info with LIR.

```
[DllImport("ril.dll")]
public static extern IntPtr RIL_GetCellTowerInfo(IntPtr
                                                   lphRil);
[DllImport("ril.dll")]
public static extern IntPtr RIL_Deinitialize(IntPtr
                                                   lphRil);
[DllImport("ril.dll")]
public static extern IntPtr RIL_GetSignalQuality(IntPtr
                                                   lphRil);
  res = RIL_GetCellTowerInfo(hRil);
  res = RIL_GetSignalQuality(hRil);


RILCELLTOWERINFO rci = new RILCELLTOWERINFO();


result += String.Format("MCC: {0}, MNC: {1}, LAC: {2},
CID: {3}, ", rci.dwMobileCountryCode,
rci.dwMobileNetworkCode, rci.dwLocationAreaCode,
rci.dwCellID);

RILSIGNALQUALITY rsq = new RILSIGNALQUALITY();


result += String.Format("Signal Quality: {0}, MinSig {1},
MaxSig {2}, LowSig {3}, HighSig {4}",
rsq.nSignalStrength, rsq.nMinSignalStrength,
rsq.nMaxSignalStrength, rsq.nLowSignalStrength,
rsq.nHighSignalStrength);
```

The GSM network provide only one BS info in each search cycle. This BS has the highest signal strength. The more BTSs info is collected by a several iteration cycles. During 10 cycles (per 10 seconds) the 4 BTS info is obtained on average.

The important info from BTSs is Signal Strength and Time Advance (TA). SS is refreshed every several seconds (in every scan) whereas TA is provided only during some type of communication with selected BTS (e.g. request to talk, move to another area - Location Area Code (LAC)). The list of these BTSs with info is further processed as in previous case for WiFi and BT networks. Only change is in usage of TA if it is accessible.

Another possibility to get the user position in outdoor space is in GPS [8]. GPS provide a position by LONgitude and LATitude (X and Y). Only simple conversion is needed to transform a GPS coordinates to S-JTSK, which is used in PDPT Framework.

2.3 The PDPT Framework Design

The PDPT framework design is based on the server-client architecture. The PDPT framework server is created as a web service to act as a bridge between MS SQL Server (other database server eventually) and PDPT PDA Clients [Fig. 5].

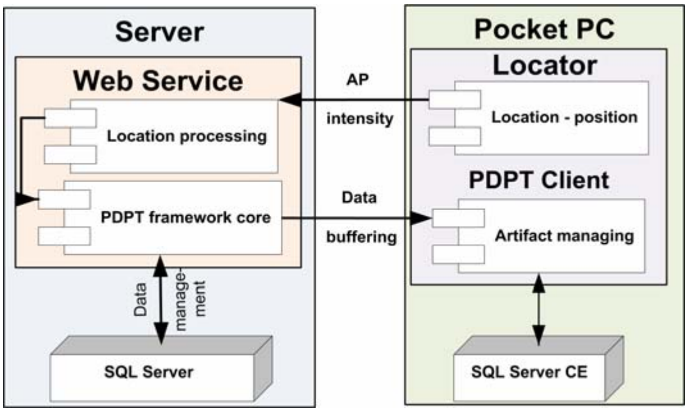


Fig. 5. PDPT architecture – UML design

Client PDA has location sensor component, which continuously sends the information about nearby AP's intensity to the PDPT Framework Core. This component processes the user's location information and it makes a decision to which part of MS SQL Server database needs to be replicated to client's SQL Server CE database [9][10]. The PDPT Core decisions constitute the most important part of PDPT framework, because the kernel must continually compute the position of the user and track, and predict his future movement. After doing this prediction the appropriate data are prebuffered to client's database for the future possible requirements. This data represent artifacts list of PDA buffer imaginary image [Fig. 6].

2.4 PDPT Core - Area Definition

The PDPT buffering and predictive PDPT buffering principle is shown in [Fig. 6]. Firstly the client must activate the PDPT on PDPT Client. This client creates a list of artifacts (PDA buffer image), which are contained in his mobile SQL Server CE database. Server create own list of artifacts (imaginary image of PDA buffer) based on area definition for actual user position and compare it with real PDA buffer image.

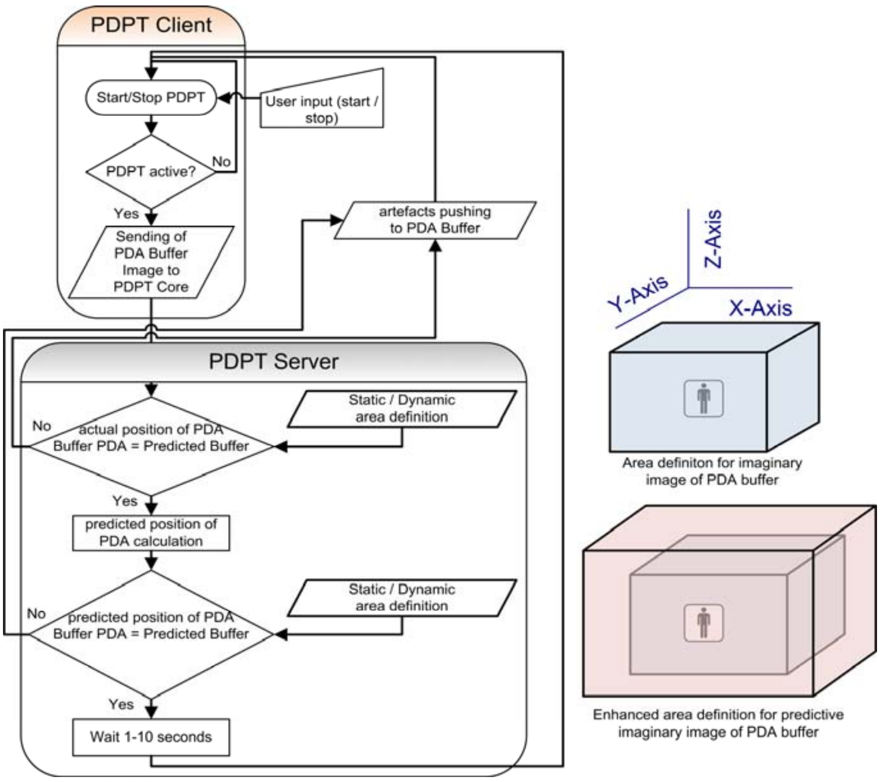
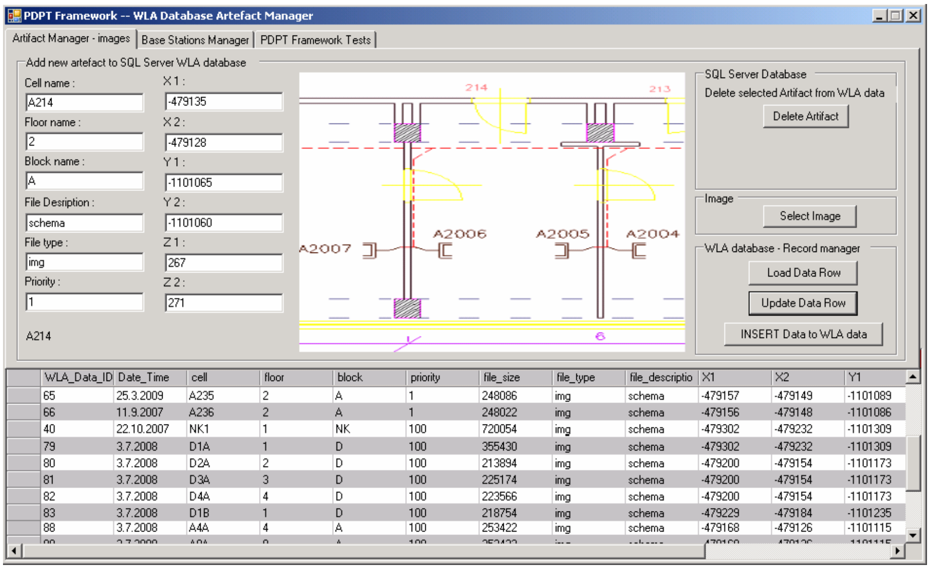

PDPT Client
Start/Stop PDPT
User input (start / stop)
PDPT active?
No
Yes
Sending of PDA Buffer Image to PDPT Core
artefacts pushing to PDA Buffer
PDPT Server
Static / Dynamic area definition
No
actual position of PDA Buffer PDA = Predicted Buffer
Yes
predicted position of PDA calculation
Static / Dynamic area definition
No
predicted position of PDA Buffer PDA = Predicted Buffer
Yes
Wait 1-10 seconds
Z-Axis
Y-Axis
X-Axis
Area definiton for imaginary image of PDA buffer
Enhanced area definition for predictive imaginary image of PDA buffer

Fig. 6. Object diagram of PDPT prebuffering and predictive PDPT prebuffering. Right part shows the area definition for imaginary image of PDA buffer.

The area can be defined as an object where the user position is in the object centre. I am using the cuboid as the object in present time for initial PDPT buffering. This cuboid has static area definition with a size of 10 x 10 x 3 (high) meters. The PDPT Core will continue with comparing of both images. In case of some difference, the rest artifacts ale prebuffered to PDA buffer. When all artifacts for current user position are in PDA buffer, there is no difference between images. In such case the PDPT Core is going to make a predicted user position. On base of this new user position it makes a new predictive enlarged imaginary image of PDA buffer. The size of this new cuboid is statically defined area of size 20 x 20 x 6 meters. The new cuboid has a center in direction of predicted user moving and includes a cuboid area for current user position. The PDPT Core compares the both new images (imaginary and real PDA buffer) and it will continue with buffering of artifacts until they are same. In real case of usage is better to create an algorithm to dynamic area definition to adapt a system to user needs more flexible in real time. For additional info please refer to [11].

2.5 PDPT Framework Data Artifact Management

The PDPT Server SQL database manages the information (artifacts) in the context of their location in building environment. This context information is same as location information about user track. The PDPT Core selecting the data to be copied from PDPT server to PDA client by context information (position info). Each database artifacts must be saved in database along the position information, to which it belongs.


PDPT Framework -- WLA Database Artefact Manager
Artifact Manager - images
Base Stations Manager
PDPT Framework Tests
Add new artefact to SQL Server WLA database
214
213
A2007
A2006
A2005
A2004
6
SQL Server Database
Delete selected Artifact from WLA data
Delete Artifact
Image
Select Image
WLA database - Record manager
Load Data Row
Update Data Row
INSERT Data to WLA data

Cell name :	X 1 :
A214	-479135
Floor name :	X 2 :
2	-479128
Block name :	Y 1 :
A	-1101065
File Desription :	Y 2 :
schema	-1101060
File type :	Z 1 :
img	267
Priority :	Z 2 :
1	271

A214

WLA_Data_ID	Date_Time	cell	floor	block	priority	file_size	file_type	file_descriptio	X1	X2	Y1
65	25.3.2009	A235	2	A	1	248086	img	schema	-479157	-479149	-1101089
66	11.9.2007	A236	2	A	1	248022	img	schema	-479156	-479148	-1101086
40	22.10.2007	NK1	1	NK	100	720054	img	schema	-479302	-479232	-1101309
79	3.7.2008	D1A	1	D	100	355430	img	schema	-479302	-479232	-1101309
80	3.7.2008	D2A	2	D	100	213894	img	schema	-479200	-479154	-1101173
81	3.7.2008	D3A	3	D	100	225174	img	schema	-479200	-479154	-1101173
82	3.7.2008	D4A	4	D	100	223566	img	schema	-479200	-479154	-1101173
83	3.7.2008	D1B	1	D	100	218754	img	schema	-479229	-479184	-1101235
88	3.7.2008	A4A	4	A	100	253422	img	schema	-479168	-479126	-1101115

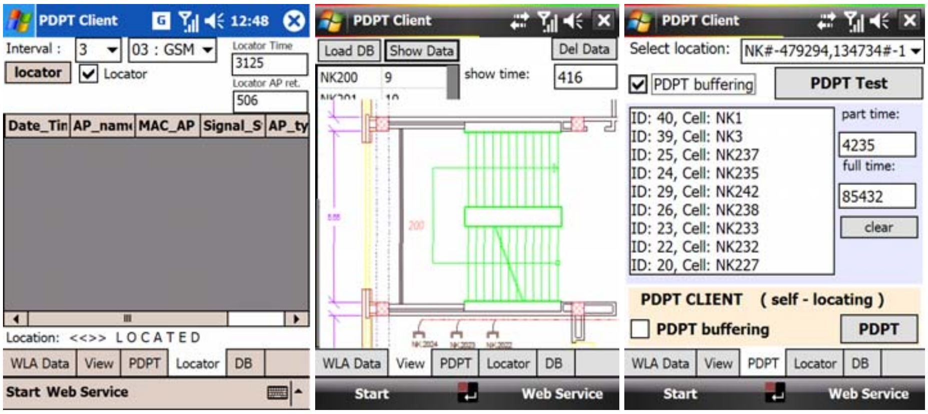
Fig. 7. PDPT Framework Data Artifact Manager

During the creating process of PDPT Framework the new software application called "Data Artifacts Manager" was created. This application manages the artifacts in WLA database (localization oriented database). User can set the priority, location, and other metadata of the artifact [Fig 7]. The Manager allows creating a new artifact from multimedia file (image, video, sound, etc.), and work with existing artifacts. More info can be found in chapter 3.1 [5].

The needs of interface to operate with APs info arose out of the developing process of PDPT Framework. The enhancement of Artifact manager was created on that ground. Now the Artifact Manager contains a new tab "Base Stations Manager" to operate with APs or BSs of selected networks [Fig. 7]. This manager is connected directly to PDPT Server database, to tables WiFi_AP, BT_AP, GSM_BTS [Fig. 3].

2.6 The PDPT Client Application

The PDPT Client application realizes thick client to the server side and an extension by PDPT and Locator modules. Figure [Fig. 8] shows three screenshots from the mobile client. The first one [Fig. 8a] shows the Locator module with selected GSM scanning. The info text box "Locator AP ret." Provide info about last founded GSM BTSs and number of recognized BTSs (BTSs with GPS position). In current case the 6 BTSs was founded and 5 of them was recognized by PDPT Framework. Figure [Fig. 8b] shows the classical view of the data artifact presentation from MS SQL CE database to user (in this case the image of Ethernet plan of the current area). The PDPT tab [Fig. 8c] presents a way to tune the settings of PDPT Framework. The middle section shows the logging info about the prebuffering process. The right side means measure the time of one artifact loading ("part time") and full time of prebuffering in millisecond resolution. More screens and details of PDPT Client can be found in chapters 2.7 and 2.8 [5].


Interval : 3
03 : GSM
Locator Time 3125
Locator AP ret. 506
☑ Locator

Date_Tin	AP_nam	MAC_AP	Signal_S	AP_ty

Location: <<>> LOCATED

NK200 9
NK201 10
show time: 416

Select location: NK#-479294,134734#-1
☑ PDPT buffering

ID: 40, Cell: NK1
ID: 39, Cell: NK3
ID: 25, Cell: NK237
ID: 24, Cell: NK235
ID: 29, Cell: NK242
ID: 26, Cell: NK238
ID: 23, Cell: NK233
ID: 22, Cell: NK232
ID: 20, Cell: NK227

part time: 4235
full time: 85432

PDPT CLIENT (self - locating)
☐ PDPT buffering

Fig. 8. PDPT Client – Left one figure 8a – Locator component with GSM scanning. Middle one figure 8b – View of classical client data representation. Right one figure 8c - The PDPT options screen allow to start and control the PDPT buffering.

3 Conclusions

I am focused on the real usage of the developed PDPT Framework on a wide range of wireless mobile devices and its main issue at increased data transfer. For testing purpose, five mobile devices were selected with different hardware and software capabilities. The high success rate found in the test data surpassed our expectations. This rate varies from 84 to 96 %. Please see the chapter 4 [5] for more info.

The PDPT prebuffering techniques can improve the using of medium or large artifacts on wireless mobile lightweight devices connected to information systems. The localization part of PDPT framework is currently used in another project of biotelemetrical system for home care named Guardian to make a patient's life safer [12]. Another utilization of PDPT consists in use of others wireless networks like BT, GSM/UMTS, WiMAX, or in GPS. This idea can be used inside the information systems like botanical or zoological gardens where the GPS navigation can be used in outdoor. The BT and GSM data collecting and processing is described in this article along with sample code. Some improvements of Locator module or Artifact Manager are described as well as improved architecture of PDPT server database. The larger area of PDPT utilization can improve importance of PDPT Framework in wireless mobile lightweight systems.

Acknowledgment. This work was supported by the Ministry of Education of the Czech Republic under Project 1M0567.

References

1. Abowd, G., Dey, A., Brown, P., et al.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, p. 304. Springer, Heidelberg (1999)
2. Krejcar, O.: User Localization for Intelligent Crisis Management. In: AIAI 2006, 3rd IFIP Conference on Artificial Intelligence Applications and Innovation, Boston, USA, pp. 221–227 (2006)
3. Krejcar, O., Cernohorsky, J.: Database Prebuffering as a Way to Create a Mobile Control and Information System with Better Response Time. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 489–498. Springer, Heidelberg (2008)
4. Krejcar, O.: PDPT Framework - Building Information System with Wireless Connected Mobile Devices. In: ICINCO 2006, 3rd International Conference on Informatics in Control, Automation and Robotics, Setubal, Portugal, August 1-5, pp. 162–167 (2006)
5. Krejcar, O., Cernohorsky, J.: New Possibilities of Intelligent Crisis Management by Large Multimedia Artifacts Prebuffering. In: I.T. Revolutions 2008, Venice, Italy, December 17-19. LNICST. Springer, Heidelberg (2008)
6. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
7. Haklay, M., Zafiri, A.: Usability engineering for GIS: learning from a screenshot. The Cartographic Journal 45(2), 87–97 (2008)
8. Evennou, F., Marx, F.: Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. In: Eurasip journal on applied signal processing, Hindawi publishing corp., New York, USA (2006)
9. Arikan, E., Jenq, J.: Microsoft SQL Server interface for mobile devices. In: Proceedings of the 4th International Conference on Cybernetics and Information Technologies, Systems and Applications/5th Int Conf on Computing, Communications and Control Technologies, Orlando, FL, USA, July 12-15 (2007)
10. Jewett, M., Lasker, S., Swigart, S.: SQL server everywhere: Just another database? Developer focused from start to finish. DR DOBBS Journal 31(12) (2006)
11. Krejcar, O.: Utilization Possibilities of Area Definition in User Space for User-Centric Pervasive-Adaptive Systems. In: First International Workshop on User-Centric Pervasive Adaptation, UCPA 2009, Berlin, Germany, April 27. LNICST. Springer, Heidelberg (2009)
12. Janckulik, D., Krejcar, O., Martinovic, J.: Personal Telemetric System – Guardian. In: Biodevices 2008, Insticc Setubal, Funchal, Portugal, pp. 170–173 (2008)