

MAP / BAP PROPOSALS

PROPOSAL 1: SENSELESS LIGHT

Although Senseless is an ambitious project to unify all WSN Technologies, it adds the difficulty that any distributed application imposes. To improve the time spent on the actual localization technologies, it would be worthwhile to *downsource* Senseless into a single application to serve the localization research and Scala integration. This application would be oriented towards a single platform, Telosb/TinyOS.

Features worthwhile investigating would be:

- A C-based serial forwarder. As opposed to the current Java serial forwarder. The problem here is that you need to run the forwarder in a Linux distribution or under Cygwin. Also troublesome is the fact that the Java version fully occupies a single core or the entire processor. This is probably due to the method of polling that the serial forwarder uses. So the objectives of this subtask is
 - To make the WSN-PC bridge independent of POSIX environments, making it possible to integrate the bridge via a wrapper in any .NET project
 - Dramatically improve the efficiency, the C-version could be wrapped in for example C# without taking up too much resources
- An overhaul of the database design: Senseless was expanded to be backwards compatible, however we feel that the framework is not yet mature enough to support this. It further imposes restrictions on any future work.

PROPOSAL 2: INTERFACES

Currently, Senseless supports XML over TCP as this is supported on almost any platform including GSMs, PDAs. However some component are never intended to be run on a PDA but rather on a fully fledged desktop. It could be worthwhile to also support a more featured software interface such as WCF. WCF also has the benefit that the underlying transport mechanism can be chosen so that more platforms can be supported. A web services interface could be interesting as well, due to its popularity. Another benefit of WCF is that it dramatically eases the development time of a distributed application, as remote object/functions can be accessed as if they were local object/functions. The current interface is fully text-based, which requires that you make no typing errors of course!

The structure of the interface needs to be reviewed as well. Currently the interface is actually more or less a wrapper for the Database stored procedure. However this limits the expandability of the interface. We would advise to reform the interface to something like the one used for Scala, which is based on the ANSI RTLS API.

PROPOSAL 3: DATABASE/OBJECT PERSISTENCE

The communication with the database happens via ODBC which is implemented with ADO.NET in our project. We also have a project which wraps this functionality in to a single important function (Query). For localization however, some of the data is kept in local objects. Unfortunately this adds some work. If the data in the database changes, it also has to be changed in the local objects. This can often be a source of confusion or programming errors. It would be interesting to add a persistence layer to the project to make sure that the database and the local objects are kept synchronized.

Furthermore an ORM (Object Relational Mapping) would be interesting. Communication with the database is done with ADO.NET, however the tables need to be addressed via SQL / Stored procedures which is text-based. Again, an object based implementation is a bonus. Technologies worth checking out might be the Castle Project (Active Record Pattern) and LINQ (Microsoft's unified method of accessing data)

PROPOSAL 4: IMPROVE RSS MODELING

One of the main difficulties in localization with RSS, is that RSS is an inaccurate ranging method if it is not modeled properly. One possibility is to account for fast fading effect for example. These can be minimized by changing the carrier frequency.

One of the things we investigated was the node orientation, we did not however incorporate a solution which builds on this experiment. The TelosB nodes could be equipped with a compass to take this into account. However one would need to consider the cost of this compass. If the cost is higher than the cost required for ToA or Ultrasound hardware then it is not worthwhile.

PROPOSAL 5: IMPROVE POSITIONING ALGORITHMS

For this master thesis we gathered an immense amount of literature and analyzed a sizeable portion of it. Future students could make use of the papers and information to build a more complex and hopefully accurate algorithm.