# A MAEMO Client for Web Services of a Trading Business System

Igor O. Semenov, Pavel Yu. Andrianov, Denis D. Zabirohin,
Ekaterina V. Zabolotskih, Dmitry G. Korzun

Petrozavodsk State University
Republic of Karelia, Russia
Email: {isemenov, andriano, zabirohi, zabolots, dkorzun}@cs.karelia.ru

**Abstract**

The project develops a demo C++ application that automates in a mobile way the work with a trading business system (TBS) and exploits the distinguished mobility potential of Nokia Internet tablets (maemo platform), service-oriented architecture (SOA) and web-services. The application allows making customer orders and sales, producing various reports, and synchronizing with TBS. Our solution can support different web-services that a user selects. Local data base is implemented using SQLite. User interface focuses on ergonomic and finger-touch properties.

We present a mobile automation solution for a wide range of trading business systems (TBS). It exploits the potential of Nokia Internet tablets (Maemo-based) and Service-Oriented Architecture (SOA). Typical TBS use cases assume extensive use and maintenance of a central database (CDB). Regardless of the location and quality of the network environment or access channel, our user can retrieve, process, and store business data, e.g., make orders for goods, checking goods availability, or viewing business reports. When a network is unavailable the local mirror database is used (synchronized with CBD when possible). This service is very useful for the trade agents that travel to customers and would benefit from making on-site sell agreements (e.g., PRE-selling and VAN-selling) and for individuals who want to sell or buy goods on equal rights as the large retailed companies (TBS implements a rendezvous service keeping seller announcements).

Nowadays there is a wide range of TBSs at the software solutions market for PCs. Examples include SAP R\3, Microsoft Axapta, and Oracle E-Business Suite. It is obvious, however, that the mobile device extensions of these solutions will focus on providing interface to the PC-like system, so likely no new business models will be available.

Our approach is to combine the abilities of classical TBS (potentially a proprietary or any well know TBS system could be used as a back-end TBS) and combine it with the idea of free access of small players to the large training markets. To explain the idea let us discuss the following two use case scenarios.

Let us take a first example scenario when TBS includes accounting module (fiscal, taxation, personnel) and support for trading, logistic, and some other business processes. The system users are trade agents, storekeepers, managers, etc. Basically in the use case scenarios, the TBS maintains a central database (CDB), and the users retrieve and store business data, view reports, and produce and consume documents. Often the business processes require extensive traveling (outside company buildings) and it would be a competitive advantage to simplify work processes if the user would have access to TBS data and could input and collect new data from clients (nowadays it is typically collected in paper form or in text files at PDA or laptops). Today's approach requires that after returning to the office, the user or somebody else updates the data in the CDB, which leads to extra work, results in mistakes when reading user's notes, and unwanted delays in the business process. There are already some mobile device oriented solutions, but they are restricted to a certain TBS and cannot give a flexibility that one could expect from solutions build on the SOA principles.

The second example scenario addressed by our project is for small or medium enterprises (SME) and even individuals who what to sell and/or buy goods having the equal infrastructural abilities as the large players. The user publishes sell announcements in the common TBS CBD of all market offers, and when some other user finds this offer to be interesting they contact directly and negotiate final deal. In this case, TBS implements a rendezvous service for an open sellers/customers community. Similar functionality (clients) are already available for Internet based auctions and trading systems, but its availability on mobile devices will provide user with better flexibility.

In accordance with the SOA concept, the interface to CDB is based on a set of services, typically web-services integrating the Simple Object Access Protocol (SOAP). Recent solutions, however, are not pure mobile. Access the company TBS may be restricted within its LAN because of security reasons. Although a protected protocol like HTTPS can be used, the focus is on Internet browsers for PCs or laptops, hence a workplace is required when a user is outside. Moreover, a lot of security threats appear because of mobile domain. Summarizing, (i) a workplace at a third-party is insecure, (ii) deploying own secure workplaces at all points of interest is expensive, (iii) a laptop as a workplace is better than PC but it is not a pure at-hand tool compared with any mobile device (weight and size barriers).

The result service implements mobile client to TBS CBD as a MAEMO application that uses the back-end web services. The service supports both example scenarios described above and the user interface for both scenarios is the same, which is achieved thanks to the fact that the underlying business processes are conceptually very close. The interface to web services is XML-based and potentially supports a wide set of web services of various TBS.