REMSSBOT technical implementation

Maarten Galle
Project Leader/ Consultant IT, EDS, The Netherlands

REMSSBOT overcomes the technical problems that arise when you need to access regional environmental data that are located at several data providers that have different technical infrastructures. The user can search for data defining a who, what, where and when selection. The result of this selection can be information about the data, a reference to those data or access to those data in a structured format. REMSSBOT is an Object Oriented toolkit implemented in Visual Wave and in Forté. The used technology is based on CORBA 2.0 for communication and is multi-platform. User interfaces are in Windows and WEB browser. REMSSBOT can be used as a toolkit for implementation in other regions.

Introduction
In this article the technical implementation of REMSSBOT will be described.
First of all a summary of the main functional requirements is listed, as well as the implication they have for technical solution.

Why REMSSBOT?
As earlier defined, REMSSBOT intends to be a solution for the problem of searching for environmental data and accessing those data. Environmental management has often had a regional basis: the catchment of a river, a sea or an area with air pollution problems. The property of environmental phenomena is that they rarely stop at administrative borders. This is in contrast with monitoring of environmental data that is normally organised on an administrative basis. So solving an environmental problem means also crossing administrative borders.

In Europe no standardisation exists in environmental monitoring systems. Although some initiatives to standardise environmental systems have been taken, in reality each organisation has its hardware and software environment and its own way of processing the data. The results of monitoring activities are stored in a database. It is the task and the challenge of a researcher to find the right databases and to collect the needed data from those databases. To make life for this researcher somewhat easier the REMSSBOT project started.

Functionality of REMSSBOT
REMSSBOT has the following functions and characteristics:
• By using REMSSBOT a user can search for data by specifying the selection criteria. These criteria are called the “Who”, “What”, “Where” and “When” criteria, the “W4” selection.
• Because databases are managed by different organisations in different countries with often different languages, multi language facilities have to be incorporated. This is important not only for the user interface, but even more for the naming of the searched data: “Oxygen” is called “Sauerstoff” in Germany and “Zuurstof” in The Netherlands. So to find all of them, the term “Oxygen” has to be translated.
• As a result of the previous item and because of the need for a standard and complete structure of terms, the CDS is based on the EEA-thesaurus.
• Several methods of searching are incorporated: hierarchical, alphabetical and graphical. The hierarchical methods shows a tree of terms and the user can step through this as through the directory structure on a PC. The alphabetical method just shows a sorted list of terms. For the “Where” criterion a graphical selection method is also available. By clicking on a map the user can select the desired area.

The result of this “W4” selection is a list of services that can be obtained, the “R-services” (REMSSBOT FV services). The R-services can be divided in three categories:
1 R-Info;
2 R-Product (both unstructured data); and
3 R-Process (structured data).

R-Info is a simple description of where the data can be found, just a reference to a person or organisation that has the data available, probably like a book on the shelf.
R-Product offers, apart from the description, a reference to the data themselves, so that they can be accessed directly. Examples are file names or a URL to a WEB site. For REMSSBOT these data are considered as unstructured: REMSSBOT considers the data as one piece (e.g. a file) without taking the contents into account.
R-Process is the most complete service of REMSSBOT. It gives access to structured data in a database, from which the required data can be selected and for which a user can define constraints.

Technical choices
The challenge of REMSSBOT is to overcome the several technical choices to give users access to the environmental information. These technical choices may concern the
following differences in the Regional Environmental Data Providers (REDP).

- Hardware platform;
- Operating system;
- Database Management System (DBMS);
- Data structure;
- Database access: SQL or a special Application Programming Interface (API);
- Communication: connected to LAN, WAN, Internet, dial up line, etc.;
- Security: some are open, others are locked behind firewalls and proxy servers.

To make the technical choices the following technology is chosen in REMSSBOT.

**Multi-platform**

To be able to access all existing data providers REMSSBOT must run on almost every existing computer, so at least on Unix, OS2, W95, NT and Mac.

**Object-Oriented technology**

We have chosen an Object-Oriented (OO) approach. Not because OO is fashionable, but because we see real advantages in the OO approach for REMSSBOT. The initial demonstrators will not be the final systems. In future REMSSBOT is expected to be applied in many other systems as a toolkit. OO is initially not cheaper or better than a classical approach, but when a system is extended or adapted, the OO approach leads to better possibilities for reuse of code and creation of building blocks which will lead to lower cost.

**Common Object Request Brokering Architecture (CORBA)**

REMSSBOT connects systems that are geographically distributed over several locations that must be connected using suitable communication protocols. For the OO approach we chose CORBA 2.0. CORBA stands for Common Object Request Brokering Architecture. CORBA is the only vendor independent standard available on the market. It is developed by OMG (Object Management Group), and is adopted by X/Open and OSF (Open Software Foundation), the two leading open systems consortia, now consolidated in "The Open Group" that work together to deliver technological innovations and widespread adoption of open systems specifications. Many implementations of CORBA exist, that means many different languages (like Smalltalk, Java, and Forté) implemented by different vendors. CORBA uses IDL (Interface Definition Language) as the interface specification between the two systems.

**WEB technology**

REMSSBOT has chosen WEB-technology to make use of the upcoming Internet and Intranet infrastructure. This also gives the possibility of making environmental data available for the general public, as it is one of the user requirements of REMSSBOT.

**Programming environment**

REMSSBOT is implemented in two programming environments. In the regions of Scheldt (The Netherlands and Flanders, Belgium) and Attica (Greece) REMSSBOT is implemented in Visual Wave, a Smalltalk implementation and DST (Distributed Smalltalk). In Piemonte (Italy), Forté is used as the programming environment.

**REMSSBOT system architecture**

The REMSSBOT system architecture is given in Figure 1. Although not all system parts are implemented in all regions, the architecture gives the general picture on which the three demonstrators are based.

The description below is based on the Scheldt demonstrator, because it contains all the parts of the architecture.

Each rectangle in the figure represents a system part, running on a computer. In practice it is possible to combine several parts on one computer.

The system parts are divided into three groups:
1. Client/ Front end;
2. Central Parts;
3. Regional Environmental Providers/ Back end.

**Client/ Front end**

The "Client/ Front end" group contains the system parts which shows the GUI:

- a Windows GUI;
- a GUI for a WWW Browser.

Both give the user control for browsing to the thesauri, making a selection, listing available products and downloading products.

**Central Parts**

The Central Parts group contains:

- the REMSSBOT HTTP Server and the Wave Server/CDS;
- the CDS Database Server;
- the CORBA Naming Service.

The REMSSBOT HTTP Server and the Wave Server/CDS serve the WWW Browser. The HTTP Server is a normal HTTP Server that communicates with the REMSSBOT Wave Server/CDS. The REMSSBOT Wave Server/CDS is a Visual...
Wave Server application, with practically the same functionality as the SIS/CDS only with a different GUI.

The CDS Database Server contains the CDS database. This database stores:

- meta information over the products which can be found by REMSSBOT;
- thesauri to browse in (trees, lists), and make selections;
- additional information to download the products.

The CORBA Naming Service contains a central list of addresses of remote accessible systems. The REDP interfaces register themselves to the CORBA Naming Service, so CDS clients can ask for the address in order to be able to send an elementary service request (E-service) to the appropriate REDP interface.

Regional Environmental Providers/Back end

The Regional Environmental Providers/Back end group contains the system parts where the actual products (environmental data) are stored:

- REDP (Regional Environmental Data Provider);
- REDP interface.

The REDP is an existing database (like DONAR or the VMM database in the Scheldt region). It has its own definitions and structures. REMSSBOT only makes these data accessible. Each different type of REDP needs a specific REDP interface.

The REDP interface is a Visual Works Application that receives E-services from the CDS and translates them to a REDP Specific Query, asks the REDP and returns the result to the CDS.

The Regional Environmental WEB site HTTP Server is a normal HTTP Server with (regional) environmental files or pages. A reference to these pages is described in the CDS and is called an R-Product.

REMSSBOT services

The structure of the REMSSBOT services is given in Figure 2.

The result of a W4 selection is a product that can be a single service, a composed service, a Remssbot-Info or a Remssbot-Product. A single service is referred to one E-service that concerns only one data provider. A composed service is referred to more than one E-service. A composed service results in data that originate from several data providers and that are combined in one table.

In the actual demonstrators composed services are not yet implemented.

Hardware implementation

To give an example of a hardware configuration, the outline of the Scheldt implementation is presented in Figure 3.

The Scheldt demonstrator is situated at three locations: Middelburg DZL, Middelburg RIKZ (The Netherlands) and Aalst (Belgium). At all three locations the used computers are connected to Local Area Networks. The connection between the two locations in Middelburg is realised by an existing Wide Area Network. The Internet makes the connection to Aalst. Also users can be connected through the Internet. The data providers are Unix systems running on HP under HPUX and on SUN running Solaris. The other machines are PCs under Windows NT.

Steps to implement REMSSBOT

At the moment three demonstrators are operational. It is the intention that REMSSBOT will be extended in these regions with new
data providers, but it is expected that REMSSBOT will also be implemented in other regions to solve similar problems there.

A complete new REMSSBOT environment

The following steps have to be taken to implement REMSSBOT in a new environment. This means setting up a complete system where only the data providers (databases) exist. The amount of work and cost is strongly dependent on the structure of the database of the existing data providers and how much the needed REDP interface differs from the already existing interfaces. Also the needed communication plays an important role in the cost aspect.

The steps to take for setting up a new REMSSBOT environment are the following:

1. Analysis phase:
   - inventory of wishes;
   - which data providers are desired.
2. Define R-Info, R-Products and R-Services.
3. Define the E-services.
4. REDP:
   - develop REDP interfaces. Choose from which existing REDP building blocks can be inherited;
   - implement each E-service on each REDP;
   - choose a database for the lookup database;
   - fill in the lookup database.
5. Choose user interface (WEB or Windows).
6. Choose DBMS for CDS database.
7. Set up the communications:
   - CORBA;
   - Naming Server;
   - HTTP Server.
8. Develop user application interfaces.

Add a new data provider

To add only an extra data provider is in general a less extensive job.

First of all, the amount of effort and cost depends strongly on the structure of the new data provider in relation to the existing connected data providers.

Second, the effort also depends on the services that need to be implemented. If no extra services need to be implemented, that means that the already existing services are sufficient for the new data provider, no extra effort is needed.

Third, it depends on the needed communication in relation to performance requirements and already available connections.

Finally, the CDS has to be updated for the data in the new data provider.

Future activities

In the future a lot of activities still have to be completed. The objectives of the REMSSBOT project are to realise three demonstrators to prove that the system architecture really works. The users can give their opinion and an evaluation process assesses the results of the project. At this moment it looks very successful. But there are still a lot of activities that are outside the scope of the project,
that need to be done to make REMSSBOT successful in the long term.

Activities to improve REMSSBOT

CDS

REMSSBOT is based on the ETC/CDS data-model version 1.0 which was released in November 1996. Since then several new versions were released. In September version 2.0 was released. It is necessary to adapt the REMSSBOT system to this version to take full advantage of further initiatives of the ETC/CDS. We need to examine to what extent the new ETC/CDS tools can be helpful.

Also a maintenance tool has to be developed to make the initial set-up and the maintenance of the CDS much easier. Now it takes a lot of manual work to maintain the CDS.

Technical improvements and maintenance

The demonstrators are developed in an incremental process. This has many advantages considering the user input, but from a technical point of view some solutions are not optimal, and some must be changed to avoid future problems or to make possible extensions feasible.

On the point of security, by using the Internet some improvements have to be made.

Also the maintenance of REMSSBOT has to be organised. Because REMSSBOT is a toolkit that is meant to be extended and adapted, the set-up of version control and maintenance is very important. This can possibly be done best by defining building blocks that are managed and maintained.

Activities related to the extension of REMSSBOT use

More data providers in the regions

The three demonstrators will be extended with new data providers, new users, new functions and connections to new applications.

More implementations

The interest in REMSSBOT is rather high. Several organisations are considering applying REMSSBOT in their system as a tool to access data.

Further reading


Lievens, E. and Gallo, P. (1996), REMSSBOT Project D03.02 Guidelines for System Architecture, REMSSBOT project.