CoDecS: A Multi Protocol Platform for Environmental Emergency Notification and Handling

Peter Gabel, Wassili Kazakos, Ralf Kramer[1] and Gerhard deVries[2]

Abstract

Emergency support systems have to provide information on environmental emergencies timely and reliable. In order to fulfil these requirements, different technical means can be used. In this paper, we describe CoDecS, the multi protocol platform for the European Commission Urgent Radiological Information Exchange (ECURIE). Technically, CoDecS is based on the programming language Java, Java Remote Method Invocation (RMI), Java Database Connectivity (JDBC), and Java Beans, ISDN as the communication backbone, and Telex as the low tech backup communication means. We present the requirements, the overall approach, important aspects of the communication, and the software architecture. One of the major highlights of CoDecS is its unique combination of Java and Telex.

1 Introduction

1.1 Motivation and Background

After the Chernobyl accident in April 1986 the European Commission was put in charge by the Council to create a network for the rapid exchange of radiological information in case of nuclear accidents, between the member states and the Commission. A language neutral format to exchange data concerning radiological events and accidents (CIS: Convention Notification and Information Structure) (CEC 1991) was defined in collaboration with the I.A.E.A. (International Atomic Energy Agency). Furthermore, software for encoding/decoding messages was developed. Each member state was assigned the obligation to report each abnormal change of radioactivity levels to the Commission, who in turn has to forward such a messages to all member states as quickly and as reliable as possible.

[1] Forschungszentrum Informatik an der Universität Karlsruhe (FZI), Haid-und-Neu-Straße 10-14, D-76131 Karlsruhe, Germany; Email: {gabel, kazakos, http://www.fzi.de/dbs/applAreas/uis.html

[2] Joint Research Center, Environment Institute, I-21020 Ispra, Italy; Email: http://java.ei.jrc.it/

The current version of this network is based on the transmission of CIS encoded messages by telex. The software is several years old and its speed and reliability are regarded as insufficient. Since also the user interface of the current Coding and Decoding System (CDS) and its stability are considered as unsatisfactory, it was decided to write completely new software (CoDecS).

The Environment Institute of the JRC defined the requirements for the CoDecS software (deVries 1995) in collaboration with the competent authorities of the EU Member States. The purpose of the software is to assist the ECURIE contact points in creating, transmitting, receiving, decoding, and managing messages in the CI Structure. The new software is implemented in Java and set on top of an ISDN network. It will replace the existing DOS-based CDS in all the contact points. Because of its experience with environmental information systems and especially with open Internet standards like the World Wide Web and Java, FZI obtained the contract from the JRC to design and implement the software.

1.2 Outline

This paper is organized as follows. In Section 2, we describe the starting point, namely the system that is in place. In Section 3, we place our work in the context of related work. The technical cornerstones that are relevant are presented in Section 4. In Section 5, we discuss the requirements. The overall approach chosen for CoDecS and the software architecture are introduced in Section 6. We conclude the paper with the first lessons learned and an outlook on the current and future work.

2 Starting Point: The Current ECURIE System (CDS)

The current ECURIE system is a Telex-based communication network between the EC and the Member States, through which radiological information can be exchanged. At present, all ECURIE Telex messages pass through the Commission’s Telex Centre in Brussels (see Figure 1). To receive priority treatment the messages are headed by a IATA Telex code, which allows for automatic recognition of ECURIE messages.

The structure used for exchanging information is the Convention Information Structure (CEC 1991). Its main characteristic is that each piece of information is labelled. This strongly reduces the amount of transmitted information and reduces language translation problems. ECURIE messages need to be encoded to CIS before transmission and need to be decoded after reception. The C.I. tructure does not only provide information on radiological measurements, but also on predicted values, site meteorological data and taken decisions (countermeasures).

Since encoding or decoding messages in a manual way is very time consuming and is likely to cause many errors, coding-decoding software (CDS) has been developed by the IAEA (decoding part) and the JRC (encoding part). The CDS software is written in C and runs under DOS. It also allows to transmit and receive messages to and from the Telex lines if an automatic Telex device is connected.

In Figure 1 the flow of a typical ECURIE message is shown: an exercise message is generated by CEC DG XI/C/1 Radioprotection, Luxembourg (1). This Telex message, sent by internal electronic mail to the Commission Telex Centre (CTC) (2), is converted to the Telex network (3) and transmitted to the Member State contact points (4). Depending on the level of the exercise an appropriate response message is returned by the Member States to CTC (5) and transmitted back to DG XI/C/1 (6). In case of a level 3 exercise, up to two encoded messages containing data are sent by each MS contact point to the CEC and from there again forwarded to each contact point.

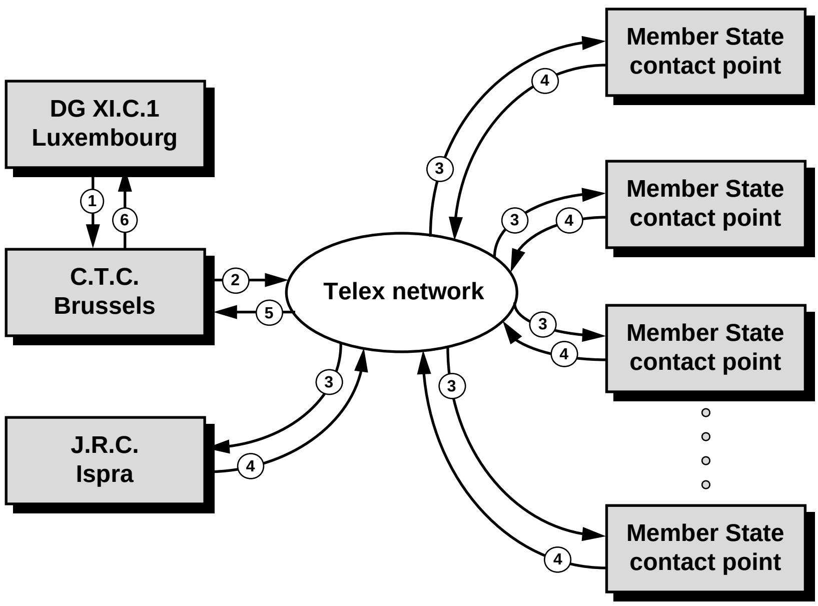

Figure 1: ECURIE message flow

3 Related Work

In this section, we place our work in the context of related areas, emergency support systems and mail systems.

We do not include systems that are based on the same technology, namely the object-oriented programming language Java, its tools and techniques, basically because of the vast area of areas it has already been used in. However, we would like state that to the best of our knowledge, CoDecS is the first and only system that combines Java technology and telex.

3.1 Emergency Support Systems

Environmental information systems can be grouped into several categories. With respect to CoDecS, two categories, namely national automatic radiological monitoring systems and transnational nuclear emergency support systems are the most closely related ones.

3.1.1 National Automatic Radiological Monitoring Systems

Many European countries have an automated radiological monitoring system. in place. The following examples are taken from the EURDEP Reference Manual and European Automatic Monitoring Systems publication (EUR 16415 EN):

- The Austrian radiation early warning and monitoring system;
- The TELERAD network for remote monitoring of the radioactivity in Belgium;
- The SWISS automatic monitoring networks for radioactivity (NADAM & MADUK);
- The integrated radiation information system in the Czech Republic (RMN);
- The German Integrated Monitoring and Information System (IMIS);
- The Spanish Environmental Surveillance Radiological Network (REVIRA);
- The Finnish Radiation Information Centre with Analysing Logic (RADICAL);
- The French national network for the radiological survey of the territory (TELERAY);
- The United Kingdom national radiation monitoring system (RIMNET);
- The national system for nuclear emergency preparedness and the automatic early warning system in Hungary (NSNEP, NERMS);
- The automatic monitoring and alarm network for the surveillance of the environmental radioactivity in Luxembourg;
- The Dutch civil defence radioactivity monitoring network (BMNI);
- The national system for monitoring gamma radiation in Norway;
- The automatic monitoring network in Poland;
- The Portuguese automatic network for the continuous monitoring of air-born radioactivity (RADNET);
- The national Environmental Radioactivity Surveillance Network in Romania (NERSN);
- The Swedish stationary Gamma monitoring stations;

- National networks of radiological monitoring in the Slovak Republic.

A common denominator of most national monitoring systems is that they continuously measure the radioactivity in the air and have a central software system that gives an alarm if values exceed a threshold. Abnormal values reported by the national monitoring system may also be the reason for the national experts to send an ECURIE initial notification to the European Commission in Luxembourg (DGXI/C/1), so that the other EC Member States will be informed (according the Euratom 87/600 treaty). ECURIE is therefore not continuously working but only activated in case of an emergency.

3.1.2 Transnational Emergency Support

Examples of nuclear emergency information and support systems that work at a transnational level (de Cort et al. 1996) are RODOS (Radio-activity On-line DecisiOn Support) (Ehrhardt et al. 1993), EURDEP (European Union Radio-activity Data Exchange Platform, both for routine and emergency cases), ECURIE (only for emergency), and IRIS (International Radioactivity monitoring and Information System).

EURDEP (EURDEP) is directly depending on the national monitoring systems, because it receives national monitoring data from most European countries and aggregates this to a European data set that is available to all the participants This daily exchange of data can be increased to a two-hourly exchange in case of a nuclear emergency. EURDEP is therefore used both for routine exchange of monitoring data and for emergency purposes.

3.2 Mail Systems

At first, CoDecS is a mail system. Hence, it is able to send and to receive messages over some electronic data communication channel. But there are fundamental differences that distinguish CoDecS from standard mail systems. CoDecS is to be used during emergency situations. That means that reliability and stability are fundamental. The normal way of transmitting messages by SMTP and storing them in the POP format would not fulfil these requirements. Therefore it was decided to transmit the ECURIE messages in a synchronous way, so that the software has the certainty that a message is delivered at the recipient station. This delivery confirmation is independent of the transmission protocols and paths and will therefore be applicable to transmissions by Telex, ISDN and public Internet. To further improve the overall reliability of the ECURIE system, CoDecS is able to use a fall-back system, so that if transmission by ISDN is for example not possible, it will try to send the message by Telex or Public Internet. The number of retries and the sequence of transmission paths can be imposed by the user.

To increase the reliability of the storage system all messages sent or received by CoDecS are stored in a relational database (Oracle Lite).

The messages sent by CoDecS use a special format, called CIS (Convention Information Structure). This CIS format allows at the same time to code the information and to reduce the size of the messages by replacing fixed text by a line number. The data that is transmitted is composed of line numbers followed by the data entered by the operator. At the receiving side, the line numbers are replaced again by the fixed text to restore the original report. This mechanism also simplifies the translation of messages in the national languages. Messages send by CoDecS can be of three different types *Initial notifications*, *Responses*, and *Notifications*. An *Initial notification* is the very first message of an accident. It will be send from the originator to the central CoDecS station in Luxembourg. From Luxembourg the *Initial notification* message will be forwarded to all member states of the EU. A response message is a message that is returned from the Member States to the EC DGXI/C/1 to indicate that the competent national authorities received the initial notification and are operative. Each Contact Point only sends a single Response message for each Initial Notification. *Notification* messages are all the further messages sent by the countries to report the further developments, protective measurements, predictions etc. Hence, as compared to standard mail systems, dedicated kinds of mails must be supported.

Standard mail systems store their data (i.e., basically the mails themselves) in the computer´s file system. Hence, special care has to be taken in order to ensure data consistency during updates. This requirement can fairly easily be fulfilled if an adequate database system can be used to store and manage the mails.

To guarantee a high level of stability and reliability, CoDecS supervises itself, the host system, and all connected devices continuously by a supervision process.

To summarise, the following features distinguish CoDecS from standard mail systems:

- guaranteed transfer between sender and receiver,
- use of arbitrary devices for transmission,
- support of language independent mails,
- support of dedicated mails only (standardised formats),
- use of a database management system, and
- automatic supervision of the CoDecS environment.

4 Basics: Java

Java (Arnold 1996) is an object-oriented programming language that has been especially designed to be used on networks of heterogeneous computers (platform independence). Java can be used both stand-alone and in conjunction with the Web. Since a detailed description of the Java programming language is beyond the scope of this paper, we focus on those basic concepts of Java which make the language so powerful and which were used for the design and implementation of CoDecS.

JavaBeans (JavaSoft 1997a) is the platform-neutral component architecture for the Java application environment. It targets the development of network-aware solutions for heterogeneous hardware and operating system environments and extends the Java capabilities to reusable component development. JavaBeans are currently used mainly for the composition of graphical user interfaces although this technology offers much more.

A remote object is an object whose methods can be invoked from objects executed under the control of a different virtual machine. Correspondingly, a remote interface defines the methods of a remote object, and a remote method invocation (RMI) is the action of invoking a remote object. It has the same syntax as the invocation of methods of local objects, i.e., objects of the same virtual machine. The Java RMI (JavaSoft 1997c) object model was designed following the object model of Java. A reference to the remote object is needed before a remote method can be invoked. A client has two possibilities to get such a reference: Either another method returns the reference, or the object uses the RMI name service.

The Java Database Connectivity (JDBC) (JavaSoft 1997b) API defines Java classes to represent database connections, SQL statements, result sets, database metadata, etc. It allows a Java programmer to issue SQL statements and process the results. JDBC is the primary API for database access in Java. The JDBC API is implemented via a driver manager that can support multiple drivers connecting to different databases. JDBC drivers can either be entirely written in Java so that they can be downloaded as part of an applet, or they can be implemented using native methods to bridge to existing database access libraries.

5 Requirements

CoDecS is going to replace the existing DOS-based CDS software so backward compatibility must be guaranteed. Its main task is to transfer CIS mails between the European Union Member States (and Switzerland) in a fast and reliable manner. Besides the generic requirement that the software has to be installed by Contact Points in all the Member States, CoDecS needs very specific functionalities to fulfil its tasks. In this section we describe these requirements one by one and indicate how they are addressed in the overall approach in CoDecS.

5.1 Reliability and Stability

One of the major requirements for CoDecS is the absolute need for reliability and stability to avoid crashes during emergencies. Memory leaks and other consumption of resources should be avoided, so that a regular restart can be avoided. In addition CoDecS should be always online to receive the notifications and use several transmission paths in parallel to guaranty the receipt and transmission of ECURIE messages. To assure this reliability a watchdog process has been implemented taking care of periodically checking the vital functions of CoDecS and of the storage and transmission devices.

5.2 Platform support

CoDecS being developed in Java will be able to run on each computer with a Java Virtual Machine (JVM). It will be extensively tested both on the Windows NT and the UNIX platform.

5.3 Communication

The CoDecS software is also used in the central node of the ECURIE network in Luxembourg and must therefore have the capability and user-friendliness to allow the operators to handle up to 1000 messages per day. Because of the nature of the messages, it must be guaranteed that each mail effectively reaches the addressee and that identical mails that arrive more than ones at a CoDecS station will be deleted automatically. Besides the Communication via TELEX, needed for the required backwards compatibility, CoDecS will use the SMTP and FTP protocols both over ISDN and public internet connections.

5.4 Database

In order to achieve persistence and to ensure that mails are not corrupted, CoDecS stores all incoming and outgoing mails in an external database. This is an Oracle database which is available for all supported platforms. The connection to the database is managed via the JDBC interface.

5.5 User Friendliness

A windows based user interface should be designed such to ensure extreme user friendliness and easy to use in emergency circumstances. The software should support users in time pressure and not familiar personnel should be assisted to assure the creation of correct messages. This includes features such as activation of pop-up windows in case of incoming mail. Besides English as the default language, CoDecS should also be able to handle one other language per station.

6 Overall Approach, Communication, and Software Architecture

6.1 Overall Approach

The heterogeneous infrastructure of the various computing environment used by participants of the ECURIE project, lead to the requirement that the CoDecS software has to run on several platforms (Windows NT, UNIX). Therefore, Java has been chosen as the programming language for the development of the CoDecS software.

The high demands on reliability and stability lead to an architecture consisting of several multi-threaded processes. The Java RMI mechanism (Remote Method Invocation) builds the background for the basic process intercommunication in a way that several processes of CoDecS simultaneously act as RMI server and clients, built up by single threads inside of these processes. For the user interface, JavaBeans with its abilities to build up independent reusable objects in a distributed environment has been chosen. With JDBC (Java Database Connectivity), standard SQL Databases can be used for storing and loading of all kind of data.

6.2 Communication Protocols

Stability and reliability are two of the most important requirements for an emergency support system. It is of crucial importance to ensure that in case of an emergency messages sent arrive at the recipient.

To fulfil these requirements two strategies are used. On the one hand, to prevent that CoDecS is unable to send messages because of a malfunction either in a transmission device or in a transmission channel, CoDecS is able to send messages over different media Telex and ISDN. On the other hand, it also allows to build up different CoDecS installations, depend on the special capabilities of a host system, for example to run CoDecS also if no telex line is available.

6.2.1 Telex

The former CDS software (see Section 2) was completely based on Telex and only able to send message over Telex lines. Mainly for that reason and because of the availability of Telex lines at all national organisations, where CoDecS will replace

CDS, the first release of CoDecS is also based on Telex as communication medium. Although Telex is not a modern communication medium, it is known as a very stable medium, which is available almost all over the world, especially in remote areas, where you can hardly find any Internet POP.

For CoDecs the problem of sending messages by Telex was to combine a modern programming language like Java with this old technology. Because of the platform independent architecture of Java a direct sending or receiving of data with the used telex device was not possible. To solve these problems, Java native interfaces were used to manage the actual sending and receiving of byte fields to and from the serial line interfaces.

6.2.2 SMTP/ISDN

The normal way to send electronic message by SMTP over the Internet, is by means of store and forward using SMTP servers until they reach their destinations. This does not suit the requirements of an immediate and acknowledged delivery of messages. In a very early phase of CoDecS, the use of X.400 as message protocol was also discussed, but later SMTP was chosen to handle a direct interconnection of remote CoDecS stations. Unlike other SMTP servers, CoDecS cannot only send messages over the public Internet. Instead it uses by preference an ISDN line as communication medium to establish a direct connection to the recipient SMTP server at a remote CoDecS station. Also in case of a transmission of messages of the public Internet a direct connection between the calling and receiving CoDecS station is made between the two SMTP servers of CoDecS, in other words, no store and forward mechanism using intermediate POP servers is used. In the CoDecS architecture the sending of messages only makes use of the handshake mechanism of SMTP, to deliver messages correctly to a remote CoDecS SMTP server. The routing of the message is organised in a way, that dependent of the recipients address and the address data, stored inside of an address book inside of CoDecS, an ISDN line will be opened and the message is then delivered directly to the remote CoDecS station. In case of incoming messages CoDecS act as an ordinary SMTP server with full support of the SMTP protocol.

CoDecS offers two ways to open ISDN lines. In the first configuration, an ISDN router is connected to a CoDecS station. The ISDN lines are opened not by CoDecS itself, but from the ISDN router. Every time CoDecS tries to open a TCP connection to a remote CoDecS station, the ISDN router opens a ISDN according to an entry inside or its routing table to the remote CoDecS station. This configuration has the advantage that ISDN routers and their routing tables can be managed and configured remotely from one central instance.

In the second configuration, CoDecS builds up the ISDN connection itself. In this configuration, an internal or external ISDN card is used instead of an ISDN router.

When sending a message, CoDecS opens itself the ISDN connection to a remote station. After establishing the ISDN line, PPP (Point-to-Point-protocol) is started. Though the management of the ISDN card cannot be completely platform independent, RAS for NT stations or PPP-daemons for common UNIX system offer both a similar ways for establishing PPP-connections over an ISDN line.

6.3 Software Architecture

The most relevant aspects for a software architecture for CoDecS is to build up a system which meets very high requirements to stability and reliability. Therefore the CoDecS architecture is build up of different specialised processes interacting with each other. All processes act both as RMI clients and servers offering its specialised services to each other. The basic processes of the system are:

- **Communication Server**: The core of the communication system organising and controlling the sending and receiving of messages. For an outgoing messages the communication server controls which transmission media have to be chosen, that means whether a message have to be sent by telex or ISDN for a given recipient address and sends the message to the correspondent interface process. In case of incoming messages all interface processes send new arrived message directly to the communication server. The communication server organises all tasks connected to an arrival of a new message like storing of the message into the database and sending of a notification to the GUI, or activation of the so-called alarm box. The communication server also polls a dedicate incoming directory and takes all files found inside of that directory as newly arrived messages. If an alarm box is connected to a CoDecS host the interaction mechanism with that external device is also part of the communication server.
- **Database process**: The database process is building up the database management system of a CoDecS system. It stores all outgoing and incoming messages into a relational database. Additionally the database process keeps control of an address book and a set of core parameters for a CoDecS installation. As relational database for the standard CoDecS system Personal Oracle Lite was chosen, but in principle it should be possible to take every relational database for which a proper JDBC driver exists.
- **GUI process**: Interface to the user. The GUI process is build up as a modern comfortable user interface, which allows a user to have an easy and fast overview over the current status of a CoDecS system. The whole GUI process is build up in a language neutral format, where all displayed text can be translated into the language used by the stuff of a CoDecS installation in any country of the European Union. That offers the possibility to use English as default language and additionally one other language per station.

- **Watchdog process**: The watchdog process has the task to control the CoDecS system itself. Every installation is checked in different time intervals, whether all parts are working correctly or whether there is a serious malfunction. The check is divided into three different parts, a *software check*, where all parts of the CoDecS system is checked, a *system check*, whether the host system is in a safe state, e.g. whether there is still enough disk space for the database system, and a *device check*, whether all connected devices, like the telex device or the alarm box, are still working correctly.

Beside of these core processes, two interface processes are part of a CoDecS system. These processes depend on the platform and the devices connected to this platform are not part of a basic installation, which is able to run also without any interface process. A interface processes deals with the sending and receiving of messages over a communication media. Dependent on whether an telex device is connected to a serial line interface or whether an ISDN is part of a host system, these interface processes can integrated into a CoDecS system.

- **Telex Process**: This process interacts with a telex device connected to a serial line interface to send or receive messages over a telex line. The telex process is build up in a way, that different telex devices can be connected to a CoDecS station. All commands sent or received to or from the telex device can be substituted by alternative command sets inside of an external parameter file.
- **SMTP Process**: The SMTP process has to deal with two different ways to send messages. At first it acts as an ordinary SMTP client/server, sending outgoing messages over a TCP connection to a remote SMTP server. To receive messages it is bound to the standard SMTP port (port 25) and incoming message from a remote station can be send directly to a CoDecS station. Message sent over an ISDN line or across the internet are handled identically. On the other hand if a message has to be send over a ISDN line, and no ISDN router is used for a CoDecS station, the SMTP process also has the task to manage the establishing of the ISDN line and of the PPP connection to the recipients host. For NT stations the RAS (Remote Access Service) mechanism, and for UNIX stations the standard PPP-daemon is used, to establish IP connections between two CoDecS hosts. These ways were chosen, because they are both standardised services and both mechanisms offer similar calling sequences to build up connections to a ISDN number and to start a PPP service to a remote host.

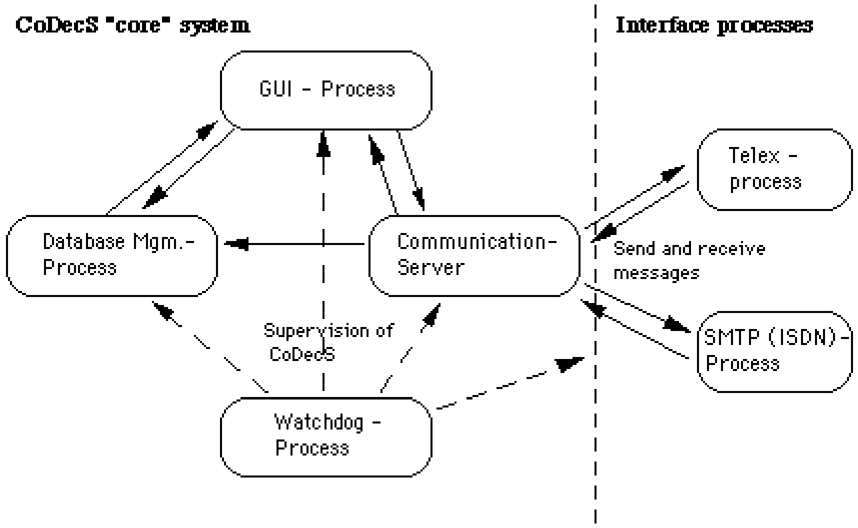

Figure 2 Processes and basic process interaction of CoDecS

Figure 2 the CoDecS processes and a general interaction mechanism are shown. A line connecting two ore more processes means there is an interaction between them, that a process uses a service offered by another process of CoDecS.

The realisation of CoDecS as a client/server system and the use of Java RMI makes it possible to place the processes of CoDecS on different machines, if necessary. This can be useful if the Oracle database is on a remote system or the telex device is connected to another machine with the ISDN card. Nevertheless all processes can be placed on a single machine also.

One general problem when using Java (JDK 1.1.x) is that Java's platform independent concept does not allow low level programming like direct access to serial line interfaces or ISDN services. To have access to hardware dependent services the direct access to send or receive data to a serial line interface these parts were realised with C++ and collected inside of a dynamic library (DLL on NT and SO-Library on UNIX stations). The JNI (Java Native Interface) mechanisms allows CoDecS to use these modules from other modules programmed with Java. Because only the actual sending and receiving of data had to be implemented in C++ it was possible to reduce the size of the C++ programming code to approximately 150 lines of code.

7 Conclusions

7.1 Lessons Learned

In this paper we described the design an implementation of CoDecS, the multi protocol platform for the European Commission Urgent Radiological Information Exchange (ECURIE). Java RMI, JDBC, multi threading, and JavaBeans were used to achieve a highly platform independent client/server and multi-tier software, the communication employs both Telex and state-of-the-art ISDN.

During the development, tools currently available for Java revealed still to be in their infancy. Hence, the development was fairly time consuming. On the other hand it was both very interesting and promising to follow the development of Java itself and of the different Java environments like development and test tools, for example to recognize a significant speed-up of program execution from JDK version 1.1.4 to JDK 1.1.6 which includes a just-in-time compiler.

An unforeseen problem was to get a new telex line. No telecommunication company, neither the Deutsche Telekom nor any of the new private telecommunication companies still offer the installation of new telex lines. As a consequence, the initial testing had to be done without having real access to a telex line.

7.2 Future Work

For the future it is also planned to integrate FTP into the CoDecS architecture. Like SMTP, FTP would primarily also be based on ISDN lines, but would probably also use the Internet as transport medium. FTP would allow to connect the file systems and databases of the remote CoDecS stations even closer to each other. Additionally it would also be possible to integrate a FTP-Server into CoDecS, such that all kind of data like measuring data or other data collected from any authorised (perhaps mobile) station connected to the internet could be easily send to a CoDecS station.

Bibliography

K. Arnold, J. Gosling, '*The Java Programming Language*', Addison-Wesley, 1996.

CEC - Convention notification and information structure, DOC No 6019/1/91 EN, 6 June 1991.

M. de Cort, H. Leeb, G. de Vries, L. Breitenbach and W. Weiss: 'International exchange of Radiological information in the event of a nuclear accident - future perspectives', The Radiological consequences of the Chernobyl accident, EUR 16544, 1996

G. de Vries, M. De Cort, S. Giuliano, S. Gatti and F. Fontana, Technical Note: 'Concept and development of an exchange format for European Radioactivity data', SP I 9638

G. de Vries: Technical Note: 'User Requirements CoDecS – ECURIE Project', 1995.

J. Ehrhardt, J. Paesler-Sauer, O. Schuele, G. Benz, M. Raffat and J. Richter: 'Development of RODOS, a Comprehensive Decision Support System for Nuclear Emergencies in European Overview', Radiation Protection Dosimetry, Vol. 50, Nos 2-4, pp 195-203 (1993).

EURDEP reference manual and European monitoring systems, EUR 16415EN

JavaSoft (1997a):'*JavaBeans Documentation*', http://java.sun.com/beans/docs/index.html, Sept. 1997.

JavaSoft (1997b):'*JDBC Documentation*', http.//java.sun.com/products/jdk/1.1/docs /guide/jdbc/index.html, April 1997.

JavaSoft (1997c) :'*RMI Documentation*', http://java.sun.com/products/jdk/1.1/docs/guide/ rmi/index.html, Sept. 1997.