Central and local survey administration through communicating data systems

Thomas Hoel
Statistics Norway

1 Overview of the system

Figure 1 shows the three main parts of the new CAI system of Statistics Norway:

- The central administrative database
- The communication system
- The interviewer laptops and their databases

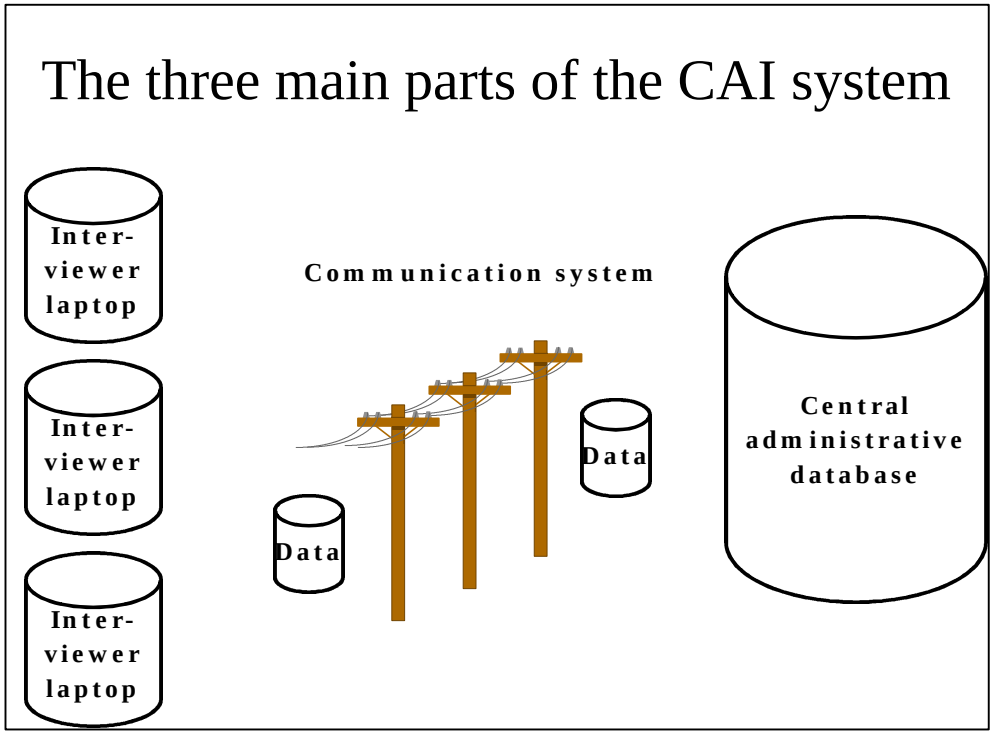

Figure 1: The three main parts of the CAI system

Because of differences in their work area, the three parts are based on different tools. Yet they fit tightly together, through defined interfaces. Table 1 below sums up work scopes, tools and data formats.

Part name	Work scope	Tools	Data formats
Central administrative database	Manages the interviewers, the questionnaires, the respondents, administrative data and questionnaire data.	Oracle and Blaise (Manipula)	Administrative data: Oracle Questionnaire data: Blaise
Communication system	Maintains the connection between the laptops and the central database. Produces a number of status reports.	Internet Explorer 5.0 and Java	Administrative data: ANSI Questionnaire data: Blaise Communication packets: XML
Laptop databases	Manages the questionnaires, the respondents, administrative data and questionnaire data.	Blaise (Manipula)	Administrative data: Blaise Questionnaire data: Blaise

Table 1: Work scopes, tools and data formats.

The term “communication system” may be a bit narrow, since the Java part of the system in addition to the communication performs a number of administrative tasks. There are, in fact, no sharp boundaries between the Java part of the system and the

Oracle part. As will become evident later in this document, the two parts overlap to some extent.

The rest of this document will examine each of the three parts in more detail.

2 The central administrative database

2.1 Overview

The central database is divided in an Oracle part and a Blaise part. The division is based on functional criteria:

- Administrative data is stored in an Oracle database in Oracle format.
- Questionnaire data is stored in Blaise databases in Blaise format.

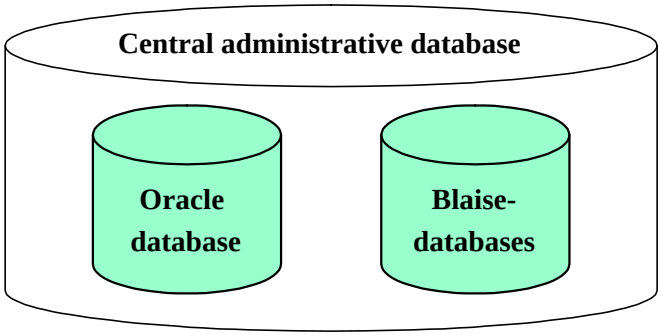

Figure 2: The main division of the central database

The reason for dividing the central database like this was tests that showed us that any conversion from or to the Blaise format led to a loss of some aspects of the data. Consistent use of the Blaise format seemed to be the only way to preserve all parts of the interviewer input: responses, comments and suppressed warnings.

The two parts of the central database is managed from the Oracle part. All events in the central database are initiated in the Oracle part. The Oracle database "knows" about the Blaise databases, knows where to find them and have methods to use them. The Blaise databases are there simply to store the questionnaire data in a convenient format and are absolutely ignorant of anything outside themselves.

From a technical point of view, it may be interesting to note that the Oracle database resides on a Unix machine, while the Blaise databases are stored on an NT file server. The user interfaces to the Oracle database runs under NT.

To get a better understanding of how the central database works, it may be useful to be familiar with some of the basic terms of the system.

2.2 Some central notions in the data model

2.2.1 Project and form

A survey **project** is an entity from the viewpoint of bookkeeping – accounts are rendered per project. A **form**, on the other hand, corresponds to a Blaise questionnaire. A survey project consists of one or more forms.

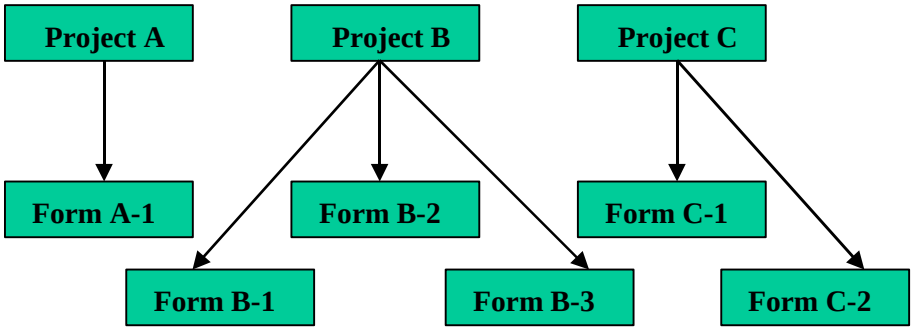

Figure 3: The hierarchy of projects and forms

2.2.2 Interview object (respondent) and period

An **interview object** is an entity that delivers data for a form (a person, a firm etc). Every interview object belongs to a form.

In many cases we want our interview objects to be dispersed over several stretches of time. For this purpose every form is subdivided into one or more **periods**, and every interview object is assigned to its form through one of the periods of the form.

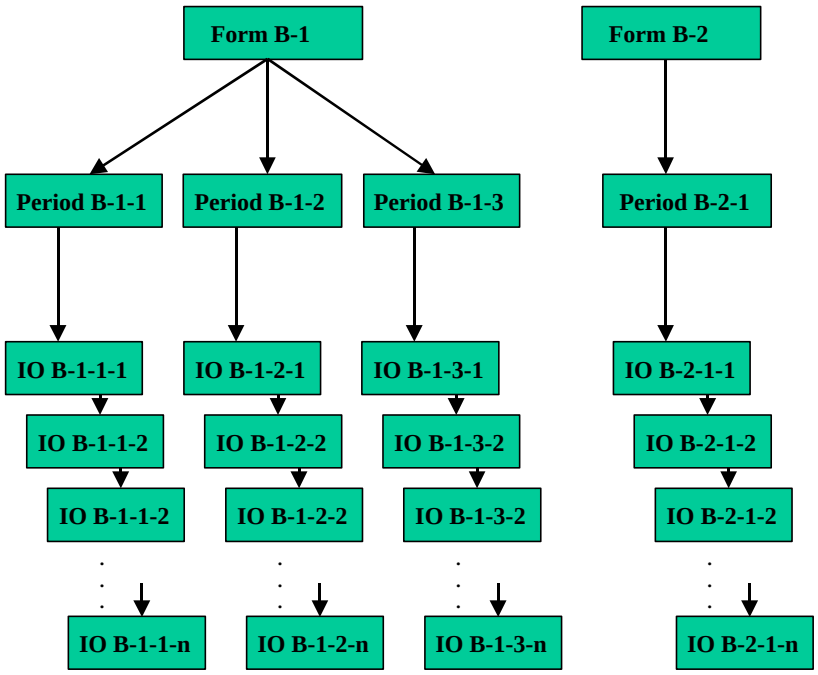

Figure 4: The hierarchy of forms, periods and interview objects

As earlier the interview objects are mostly sampled from the central Norwegian population register, or alternatively from the central industry register. The sampling is carried out in a separate system, not part of the CAI system. After sampling and necessary customization, the samples are loaded into the database.

2.2.3 Task and interviewer

An interview object prepared for a contact approach is called a **task**. An **interviewer** is an entity that accepts interview objects as work tasks. Defined in this way the notion **interviewer** resembles an address, and in fact an interviewer in the CAI system is not necessarily a person. Any mechanism that can accept a task and return it in a well-defined way, can act as an interviewer within the system.

This abstraction gives us a very useful degree of freedom. It is for instance possible to define one of the computers in our offices as an interviewer and then use it for interview work after regular office hours. Or we could establish a regular CATI system based on Blaise as an interviewer and let this system accept interview tasks from the CAI system.

2.2.4 Package

The communication between the central database and the interviewer laptops is for the most part handled through **packages**. There are several types of packages. The communication system knows the different types of packages and handles them accordingly. The packages are, however, not objects in the OOP sense of the word. They have no properties and generally no methods.

The most common package type is the communication container for one task. Tasks are transferred to the interviewers as packages, and they are returned in the same way. There is one Oracle table for outgoing tasks, and another for incoming ones. These two tables constitute part of the interface between the central database and the communication system.

Another type of packages contains the data model for a Blaise questionnaire. These packages are never returned to the central part of the system.

In addition to the task and data model packages, there is defined an “open” type of packages which serve system purposes. It is, for instance, possible to get an overview of the data situation (folders and files) on the interviewer laptops through one of these “open” packages, or corrupt files can be returned for remedy and later reinstallation.

2.2.5 CAP – Computer Assisted Payment

In addition to the interview data the interviewers deliver their time lists as data files. The time lists are compiled through a Blaise questionnaire, then converted to ANSI files and returned as packages. When received by the central database the CAP data is directed to a separate system for interviewer wages, which checks the data and in turn forwards it to the central governmental wages system.

2.3 Data structure

To manage the entities introduced in the text above, we use an Oracle datamodel with a design as shown in figure 5.

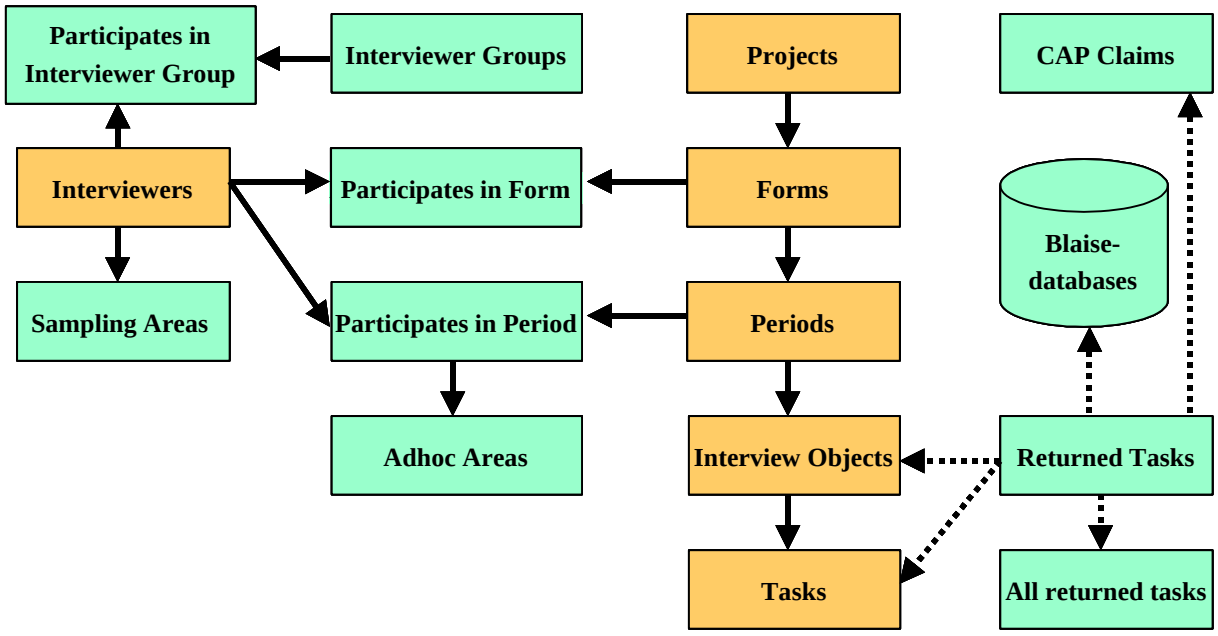

Figure 5: Data model for the central administrative database

The datamodel is mainly operated by screen applications made by Oracle Designer or Oracle Developer. The communication system has access to the database through JDBC (Java Database Connectivity), and contributes to the management with a number of useful status reports. The Oracle tools may be the safest and most flexible ones for editing the Oracle database, but when it comes to reports, Java competes very well. Reports are produced using a Java Servlet based framework resulting in ordinary HTML viewable in a web browser.

3 The communication part

3.1 Overview

The role of the communication system is to connect the interviewers to the central part of the system, transfer data between the central database and the interviewer laptops, and execute a number of defined "business methods" on the laptops based on data in the central database.

The data transfer is based on defined interfaces for collecting and delivering data. The transfer comprises a number of different types of data:

- Setups for Blaise questionnaires to be installed on the laptops
- Interview objects to be installed in the Blaise questionnaires or returned to the central database after interviewing.
- Other data files to be installed on the laptops, for instance revised versions of Manipula scripts.
- Various data to be returned to the central administration system, for instance snapshots of the contents of a laptop.
- A number of tables and status reports for the interviewers generated from the central database.

The "business methods" are predefined actions that the communication system can execute. They include:

- Installing pending questionnaires and interview objects on the laptops
- Finalize and/or remove a Blaise questionnaire from the laptops

- So called "system commands" to be executed on the laptops. A system command is a command package with a defined start method. In principle there are few limits to what you can achieve with a system command.

3.2 Security issues and internal working

Security has been a major issue in the construction of the communication system. For the same reason we cannot describe it in all details, but we can disclose that the security in relation to the central database is maintained through an internal structure containing a number of communication zones and firewalls. From the outside it is not possible to log on directly to the core of the system. All connection to the central database is initiated from the inside of the system. This mechanism slows the system down a little bit, but the capacity is still more than high enough for our 150 interviewers. The interviewers are identified by their username, password and by dial back from the central ISDN-router.

The laptops connect to the central database through an ISDN-router at their home at a speed of 64 Kb/s. The communication system is programmed in Java, and data is transferred as XML-documents. All communication is encrypted.

Ordinary internet protocols like HTTP and RMI (Java Remote Method Invocation) are used for the communication system. The communication module is implemented as a Java applet. As a consequence, the interviewers use Internet Explorer as their interface to the system. An advantage of this choice is that many interviewers know the user interface from earlier experience with computers. Also, the internet approach makes the communication system easy to maintain in the future. By using a Java applet, the communication module can be upgraded on the server, and immediately be available to all the interviewers, without upgrades being necessary on the laptops. The usual trouble with long loading times of applets is amended by applet caching, a feature of the Sun Java VM Plugin.

The interviewers all have their own homepage, presenting the status of their active projects. The homepage has links to static information like postage rates, training material and laws and regulations, as well as more dynamic reports on CAP claims.

At an early stage of the project we considered using Internet Explorer as a general interface to all programs on the laptops, which would give us the possibility to more or less hide the operating system from the interviewers. It turned out, however, that the Blaise-programs on the laptops for some reason did not function well when started from Internet Explorer, and this idea was abandoned. The main user interface on the laptops is Windows NT version 4.0. From Windows NT the interviewers may start the Blaise-applications, the communication system and some other programs.

4 What happens on the laptops

4.1 Overview

On the laptops there is a two level database structure. The top-level database contains one record for each interview object on that particular laptop. The purpose of the top-level database is to present to the interviewer a combined list of all the interview objects, no matter what questionnaire they belong to. In addition appointments with the interview objects are handled in the top-level database.

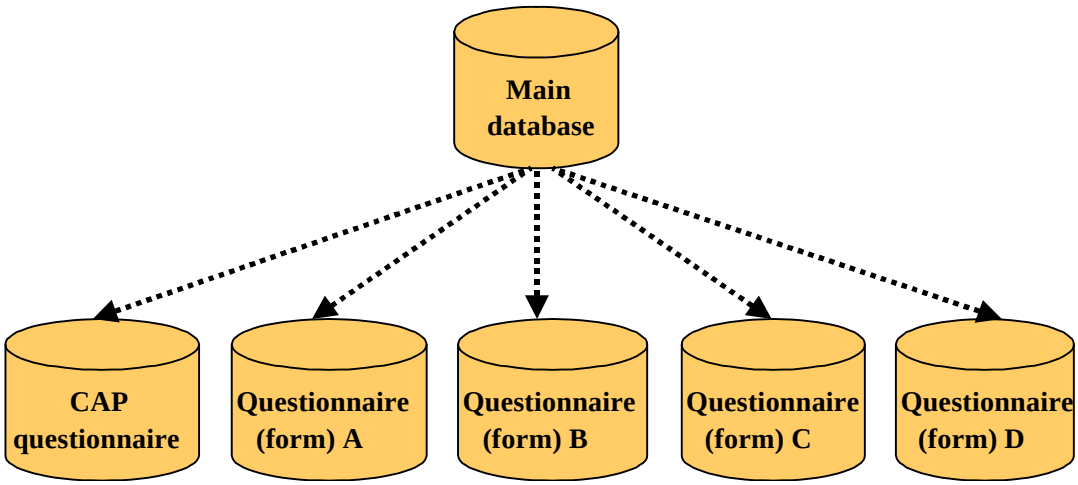

Figure 6: Database structure on the laptops

The databases on the second level are the Blaise questionnaires. Several questionnaires can exist at the same time at the second level. The relationship between the top-level database and the questionnaires on the second level is purely logical - there are no constraints or other formal structures connecting the two levels.

4.2 Management of the databases

Both the main database at the top level and the questionnaire databases at the second level are managed by a Manipula-application. For running the application we use Manipula version 4.3. The Manipula-application starts by showing the user a list of all the interviewable objects from all the accessible questionnaires. From this list an interview object can be selected for a contact approach. A contact approach can terminate with one of three statuses: Interview, non response or unfinished. After an interview or a non response the interview object is removed from the list. Unfinished interview objects remain on the list.

An interview object is transferred from the central database to the laptops in a data package, one interview object in each package. When a package is delivered to a laptop, its contents are appended both to the main database and the questionnaire database. This act of appending is done in a transaction-like way: Either both appends succeed or both are rejected and rolled back.

4.3 Packages – interface and carrier

Packages are a central part of the CAI system, and there are even two levels of packages. On the lowest level are the packages that the central database and the laptops use to communicate with each other. These packages are common zip-files, and their contents are the entities that need to be transferred between the laptops and the central database. On a higher level are the XML-packages (XML-documents) which the communication system uses to perform its part. The XML-packages resemble objects, in the OOP sense of the term, and one XML-package may contain one or more of the lower-level packages. Four types of XML-packages are defined.

At the present time five types of lower-level (zip-file) packages are known by the system. The communication system recognizes the difference between these package types, but has no knowledge of their internal structure. The role of the communication system is to collect packages at certain points in the system, deliver them at other points, and for some packages at some of the delivery points start a predefined action. The five types of packages contain:

- Setups for Blaise questionnaires to be installed on the laptops
- Interview objects to be installed in the Blaise questionnaires or returned to the central database after interviewing.
- CAP data (the interviewers' time lists) to be returned to the central database
- System commands to be executed on the laptops (program and necessities in one package)
- Various data to be returned to the central administration system, for instance snapshots of the contents of a laptop.

A short discussion of two of the package types will illustrate how the system works.

A questionnaire is installed on a laptop through a package that contains the datamodel for the questionnaire, three files in the simplest cases, plus three Manipula-scripts to handle the questionnaire. The communiaction system collects the package in the central database and transfers it to the laptop. Some folders are created for the questionnaire, and the compressed files are unzipped from the package. Now the datamodel is ready for receiving interview objects.

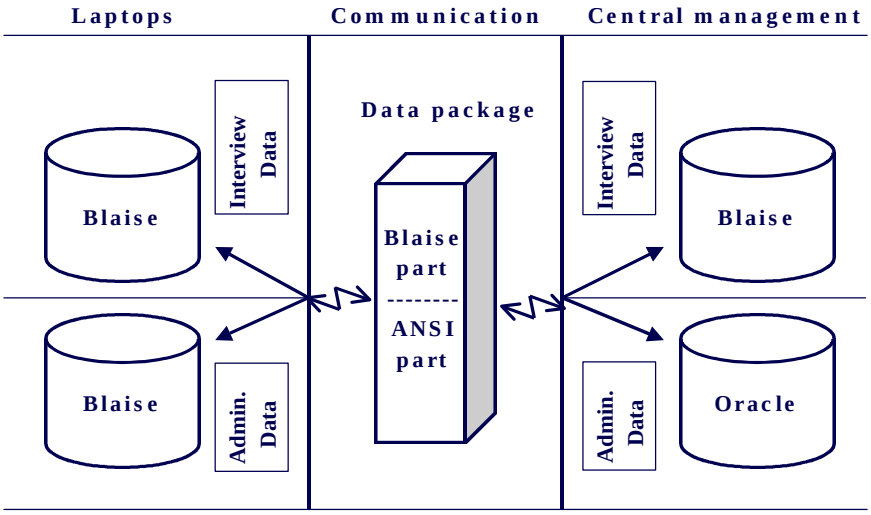

Figure 7: Work areas and data formats

The packages containing interview objects are a bit more sophisticated, as you may see from figure 7. Firstly they contain an initialized Blaise-record for the interview object. In addition they contain a file with a record of administrative data in ANSI-format for the top-level database. A new interview object is to be installed both in the top-level database and in its Blaise questionnaire. This double installation is handled by the communication system through a Manipula-script. The communication system transfers the package to the interviewer laptop, saves the package on the hard disk as a zip-file, unzips it and then lets Manipula execute the installation script.

5 A summary of a typical interview project

In the new CAI system a typical interview project will be conducted like this:

1) The Division for Sample Surveys decides to run a survey project. A **project** instance is created.
2) A **form** instance is created and a Blaise questionnaire is written.
3) An installation package is made for the form.
4) The total contact period for the form is divided into one or more **periods**.
5) A sample of **interview objects** is selected (outside the CAI system)
6) The interview objects are distributed over the periods of the form, and the sample is loaded into the database.
7) **Interviewers** are selected for each period of the form.
8) The interview objects for a period are assigned to the interviewers selected for the period (by a program).
9) **Installation packages** for the interview objects are generated.
10) The form is "opened" for use by the interviewers.
11) The interviewers collect their installation packages and install the form and their interview objects on their laptops.
12) Completed interviews and non responses are returned to the central database when the interviewers log on through the communication system. The updated status in relation to the forms in which the interviewers participate, is accessible to each interviewer on his or her homepage.
13) In the central offices the situation of the survey project can be continuously monitored through status reports from the central database.
14) When the contact period of the questionnaire is expired, the data from uncompleted interviews is returned. Then the questionnaire is removed from the interviewer laptops.
15) The questionnaire data is extracted from the Blaise database and delivered to the users.