



Concept and Structure

ProMISe is an Internet based application for the Design, Maintenance and Use of (clinical) data management projects. Using ProMISe the design and implementation of single and multi center data management projects can be performed without any programming effort, thus allowing the project coordinators as well as the users to fully concentrate on the clinical contents and logistics of the study at hand.

In the way a word processor is a generic tool to produce an unlimited variety of text documents, ProMISe is a generic program (or rather, a collection of applications) which allows a Designer to create and maintain an unlimited number of data management projects. In contrast to a word processor (like Microsoft Word) however, the source code of ProMISe is available to the Administrator and consists mainly of a collection of standard HTML pages and a Microsoft Access program for the maintenance of the implementation of ProMISe on the Server.

Although ProMISe is designed as a system which is inherently independent of the Operating System, the underlying Data base System and the supporting Web Server, the current implementation of ProMISe (both the current version 1 and the next version 2) require the following environment for the Server PC: [WindowsNT/Server or Windows2000/Advanced Server] + [SQL-Server-7] + [Internet Information Server (IIS)]. There are no special requirements to these three software implementations.

The ProMISe software furthermore requires a specific ISAPI DLL (HEITML) to be installed as an ISAPI extension to IIS. This is a DLL which processes a browser request for a certain type of webpage (a *.HEI page in contrast to a *.HTML page) and returns a normal HTML page to IIS. This is completely analogous to the functioning of Active Server Pages (*.ASP) which are processed also by a DLL. HEITML¹ is in fact an extension to HTML and provides web pages with the capability of communicating via ODBC to a data base system.

Finally the Server should have a dedicated ODBC driver to allow the HEITML DLL to communicate with the actual data base system, SQL Server. This ODBC driver is a native part of the Operating System.

You can think of ProMISe as a set of Web pages. These Web pages are both “intelligent” and “dynamic”. The content of these web pages (which are transparent to the owner of a ProMISe implementation) are a mixture of plain HTML and JavaScript and the extension of HTML called HEITML. This extension allows a richer structure of the HTML page which is very intuitive and actually “adds” some syntax like “if ... then”, “while ...endwhile” etc. to the web page. Moreover the language allows calls to data bases and the result sets can be used to build the actual web page presented to the user. The *.HEI pages behave therefore more or less as a “macro” which generates a perfectly normal, plain HTML page which is sent by IIS back to the user. The content of such a page however is dynamically determined by the nature of the user request and the content of the data base at the moment of the request. For those familiar with ASP pages: exactly the same functionality (though far more programmer friendly).

From the viewpoint of a user of ProMISe this is however irrelevant. A user does not need to know how ProMISe works since the user is simply presented with a series of webpages (a very limited set of them but each time appearing different to the user). The user can log on to a project (see below), enter some data, retrieve some information, generate a report, make a backup, perform a simple statistical analysis etc. This all happens within the completely familiar environment of normal web pages with links, buttons, tables, graphics etc.

From an organisational point of view, the following type of persons are involved in a data management project: a system administrator, a project designer and one or more project users.

The system administrator is responsible for the maintenance of the ProMISe software as such (but he/she already has the task of maintaining the entire Server infra structure anyhow). The administrator obviously needs full administrator

¹ HEITML has to be purchased separately but is extremely cheap. It does not in any respect interfere with the normal functioning of the Server.

rights to be able to install for example updates to the ProMISe system, upgrade all projects present (if necessary) and the overall backup procedures related to the Server as such but the administrator need not be involved in the maintenance of the data within the various projects.

For each project running under ProMISe there must be a Designer. This is someone with full access rights to a particular project within ProMISe and who is responsible for the structure, the Dictionary of the project itself. This person is the one who will “translate” a clinical protocol, a paper coding form into an electronic coding form and thus into an internet based data management project. From the viewpoint of ProMISe the Designer is a user with special access rights.

Normal users are associated with centers and registries. A center is a unit to which certain records in the data base belong. Usually these centers coincide with hospitals or departments therein. Users belong to such a center and each of them has its own user name and password; associated with this user name are certain access rights which can vary from “statistical overview only” to “full data management access”. A Registry is a kind of “virtual” center. Registries are inherent properties of the concepts realized in ProMISe. A Registry is actually a “view” on the data tables in the data base. These views can be defined dynamically by the Designer of a project. Hence a Registry could for example be the union of a number of centers (usually called “national registries”) or a certain subset of patients (e.g. according to a disease classification). A Registry could also be defined as a specific subset of patient for the duration of an analysis of part of an ongoing registration project. Registries can be created at will.

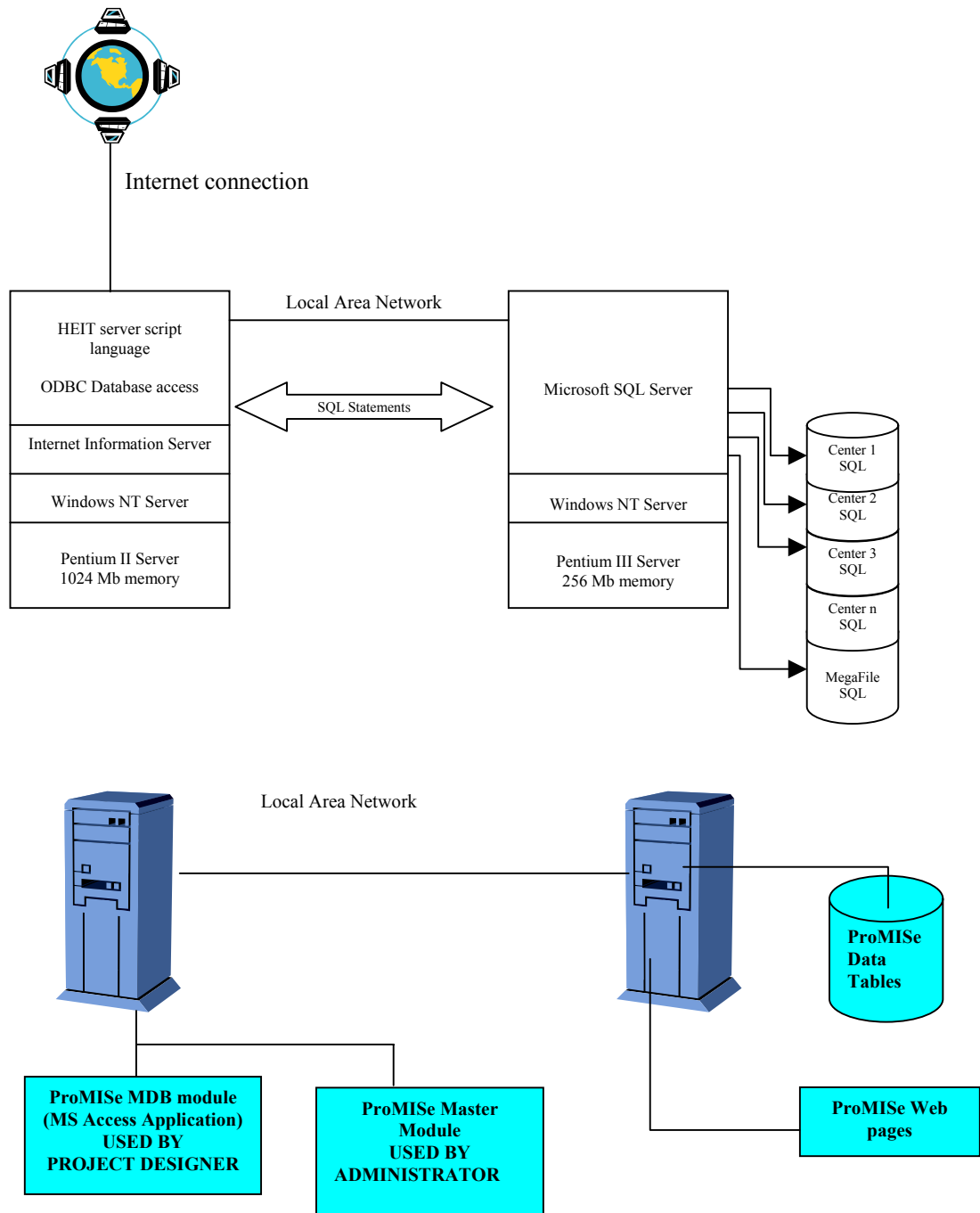
Now for the organisation of a project itself. First it should be emphasized that the entire process of designing, activating and maintaining a data management project in ProMISe does not require a single line of programming. ProMISe is a computer program which writes computer programs! The Designer of a project merely provides, in a systematic and menu driven way, clinical and structural information to the design part of the ProMISe software. On the basis of the information provided (we call this the Dictionary of a project), the ProMISe design modules then creates the entire infrastructure of the project (that is all SQL Server tables, indexes, content, web pages etc etc) within a few minutes. In this way it is possible to have a complete multi center study running everywhere in the world in a few days.....

Each project in ProMISe corresponds to one single data base in SQL Server. This data base contains a number of tables. A certain number of those tables together constitute the Dictionary of a project: these tables have a fixed structure (imposed by the ProMISe software) and their contents fully describes the structure of the project, that is the Dictionary, the clinical knowledge base. The remaining tables are truly data tables, just as in any data management project. Their fields are the data fields to be collected and their records correspond for example to patients and their follow up. The Dictionary in fact tells ProMISe in a structured way how to interpret the data tables!

In the Dictionary we find a table whose records actually describe the data tables: for each data table there is one record in this “forms table”. Then there is a table whose records actually index all the data fields in all the data tables: here we find a description of each field; the nature of its contents; its minimum and/or maximum; the way the field is coded and other administrative info. There are as many records in this table as there are fields in the data table. Then we have a table containing the coding system, that is the codes allowed for each data field and the corresponding labels. Each record describes a code in a certain labelset: for example the labelset coding “gender” might records “1=male”, “2=female” and 9=“unknown”. The actual contents is of course up to the Designer of a project. We furthermore have tables describing the quality control checks in the project; the users for a project; the centers involved in a project; the user defined queries for a project etc etc.

The Data Tables contain the actual data; together the data tables make up a relational data base. The difference with a (manually created and maintained) data base is that the exact relational structure is not created and fixed using say SQL-Server utilities but that structure is contained within the project Dictionary. A ProMISe project is self-maintaining. To explain this, consider the situation that you are running a project in full production. Then a new option has to be added to an item containing a subclassification for a disease; the only thing which happens is that the Designer adds a new record to the labelset table (through a menu: even the Designer does not work directly with the underlying table) adding a new combination of code + label (say 12=chronic leukemia). From that moment on all users of the project are presented with this additional option automatically. Likewise, suppose an additional question has to be asked in a quality-of-life questionnaire: the Designer types in the question and the possible answers (and, if necessary, logical conditions to verify the correctness of the answer by referring to the answers on other questions) and ProMISe fully automatically modifies the data table by adding the field which will contain this question. A split-second later all users are presented with this question.

In general, any change to a project is reflected immediately in the web pages and output the user obtains via the Internet. This is a direct consequence of the (crucial) design aspect of ProMISe: all web pages that together appear to the user as a data management application, do not really “exist”: they are simply generated in real time at the very moment of the request!



The figure above gives a sketch of the architecture of a ProMISe implementation. There is a crucial difference between the implementation of version 1, the version which is now in production for 2 years at the Dpt of Medical Statistics at the Leiden University Medical Center. Apart from some other experimental installations, this is currently the only full production installation of ProMISe. One should note that theoretically speaking there needs to be only one installation of ProMISe on the Internet: ProMISe itself supports an unlimited number of projects. But obviously the capacity of a single Server would never be enough when the number of projects starts growing; moreover companies or large organizations may prefer, for technical or other reasons, to run their own system.

Version 2 of ProMISe, due summer 2002, will be available to run on other Servers than the one at the Dpt of Medical Statistics. The reason why version 1 is not, is mainly logistical but also has to do with the Design Module of ProMISe. Emphasis during the development of ProMISe has been placed on the user interface and the target to have a system running for a few years in full production before attempting to design a new version that could completely independently run on other Servers without support of the creator of ProMISe and optimized on the basis of user experience. In version 1, the Designer has to use a separate Design program (menu driven but running in a DOS-Window) which creates all necessary info to generate a ProMISe project. The output of this program (the Designer does not need to know anything about programming: it is just a menu-driven application) is then converted by a Microsoft Access Application module to SQL-Server. So the design is actually a 2-stage process: typing in all the questions and answers of a paper coding form into the design program; exporting the info to tables; converting them from MS-Access tables to SQL-Server tables.

In version 2 of ProMISe this design-only part of ProMISe will also be implemented as a series of web pages. The advantage is a more uniform interface for the designer-user but, more importantly, the design takes place via a normal http: or https: connection and hence the designer can even be located outside the firewall usually surrounding the actual Server. In other words, in version 2 the Designer of an individual project does not need physical access (like a netbios drive mapping) to the Server anymore. Currently, in version 1, the Designer of a project needs to be located on the same intranet as the Server and needs to be able to perform drive mappings to the Server.

So in summary a ProMISe project is just a collection of tables in SQL Server. Some tables contain data and others contain information about these data and the structure of the project. All tables are created and maintained automatically by the "design part" of ProMISe, a set of HTML pages (and, in version 1 only, a stand-alone design program). It should be emphasized that this architecture is completely transparent and open to other applications as well. For instance a website could be allowed to extract statistical info from one of the data tables of a ProMISe project while ProMISe is completely unaware of this. An application of this could be a patient information website which automatically displays a table with the current number of transplantations having been performed for a set of diseases. Such a web page would typically be an ASP page on the same Server as the ProMISe project is running, linked to by a link on the patient site. This ASP page then contains a password to access the ProMISe project² and is allowed only to make one specific query to obtain statistical info on one particular field.

Actually, once created a ProMISe project, considered as a relational data base in SQL Server, is completely independent of ProMISe: it is just a set of tables. An organization could equally well use its own set of web pages to access the data in those tables within any programming language that is supported in the Windows-IIS environment using ODBC connections! Of course this would be ineffective as soon as changes to the structure occur, since then one would be forced to adjust these pages manually. But it is very well conceivable that a large organization would create additional pages (in addition to the standard access offered by ProMISe) for a specific project for a specific purpose, like the transmission of machine-specific data from laboratory equipment that needs to be stored in the data tables but which should not be typed in manually.

So the User part of ProMISe is just a series of interactive web pages which automatically adjust to the design and content of a specific project and which offer the data manager the functionality required in most clinical research projects. Both for randomized trials and ongoing registration projects, the built-in possibilities of this user interface should suffice for the majority of tasks to be executed. Before giving more detailed info on the options offered by the user interface, first some examples of projects currently (beginning of 2002) running in the Dpt. Of Medical Statistics.

² Since the ASP page is physically located on the ProMISe Server, this is not a security risk!

- The European Blood & Marrow Transplantation group: 500 hospitals and around 3000 members. The application covers the field of hemato-oncology and collects information on the sequelae of bone marrow transplantation in Europe in over 10 different diseases using about 2000 variables. The data base contains over 160.000 transplants and nearly 300.000 follow-up records. Apart from access by all individual hospitals (some of them enter data themselves, others send in data on paper which are entered via ProMISe on a central location by the central Registry) there are over 10 disease related Registries (to them the data base appears as containing only one specific disease) and a number of National Registries (Registries dynamically defined as the collection of patients in centers in a specific country)³.
- Within the same organization, projects are in production dedicated to:
 - Membership administration of all 3000 hospitals and members⁴
 - Administration of on-going research projects and trials in chronic leukemia
 - A randomized clinical trial in CLL.⁵
- Registration of protocols and the discussions about them in a Medical Ethical Committee. A 1-sheet coding form contains all administrative aspects of a protocol as well as a reflection of important issues during the discussions about the protocol. This administration can then be used for a yearly report as well as daily administrative tasks
- The Dutch Knee Society: a permanent registration project for 11 Dutch orthopedic surgeons collection data on knee operations. Each member has access only to his own data in detail but all can have a statistical overview (frequency tables, graphs) on the totality of all Dutch Data.⁶
- Comprehensive Cancer Centers: a clinical trial is in production and an experimental administration of all patients in all trials is being setup.
- Several trials and registration projects in hematology, orthopedics, vascular surgery.

From the viewpoint of security and reliability, the greatest advantage of the design of ProMISe is its vast user base. Whenever a bug is detected by a user, removal of that bug applies to all projects run under ProMISe. Since there is (currently) only one copy of the software, reports of users on bugs usually are implemented within a few hours or at least on the same day if possible.

Moreover, the system is continuously growing, based on the actual needs of data manager all over Europe. If for example a specific kind of report needs to be implemented for a project A⁷, this addition to the core of ProMISe is paid for by the organization running A (or by the ProMISe designer himself) is immediately available to all users of all ProMISe projects anywhere. It is as if the users of ProMISe continuously receive an updated version without any costs.

The initial design of ProMISe has been reviewed (on request by the EBMT which triggered the creation of ProMISe) by Cap Gemini Europe. This company has reviewed the basic design of ProMISe, its concepts and its security implementation. It qualified (version 0) as “state of the art” and gave a positive advice to the EBMT Board to use ProMISe as its data management system for all transplantations and clinical trials in Europe.

³ ProMISe in this way can supply a country with a truly National Registration and Information system on a specific subject by simply creating a “view”, a logical subset on the total data base. Creation of a complete national project takes about 60 seconds.

⁴ Of course a ProMISe project can handle other things than patient data; everything that can be collected on a old-fashioned plain paper coding form can be converted to a ProMISe project. My own wine cellar is an example of this.

⁵ In this particular case, randomization takes place in one country and the sequelae are administratively handled in another country: but this is no problem in an internet based environment

⁶ An important application can be a peer review system which is a natural way of using a ProMISe multi center project!

⁷ Of course this kind of report should be sufficiently general that its structure can be implemented irrespective of the actual clinical contents of the originating project; that is, it must be sufficiently generalizable to be of use to all ProMISe users.

Properties of any ProMISe project from the viewpoint of the data manager (user).

- Data Entry forms
 - Dynamically created on the basis of the current Dictionary
 - Any subset of items can be viewed as a data entry form (even user defined)
 - Menu guided data entry (possible with keyboard only to avoid RSI)
 - Unlimited quality control during data entry, based on logical criteria
 - Paper forms can be created, matching exactly the data input electronic counterparts
 - Statistical and Listing queries can be executed during data input for verification purposes

Help files can be added at liberty by the project Designer. ProMISe automatically detects help files of a *.GIF, *.DOC and *.HTM type and displays them as soon as the user needs info on a specific item. Help files can be uploaded from within a ProMISe session by the Designer.

- Reports
 - Columnar listings with values and/or labels
 - Individual reports
 - Statistical reports
 - Missing data reports
 - Survival Analyses⁸
 - Log file reports⁹

All reports can be formatted as web pages (so they can be printed using the browser standard printing functionality and as Microsoft Access data bases or Excel spreadsheets. The latter two can either be e-mailed to a user (zipped and encrypted) or downloaded by the user via an automatically created temporary link on a page during the users session).

All reports automatically generate the required SQL syntax to obtain the results asked for by the user. The mechanism is generic so queries over multiple tables appear as simple to the user as queries over 1 table. Numerous options are available to obtain the desired contents and layout of the reports. All formatting essentially takes place in the browser window

- Administrative information
 - Address data base of participating centers¹⁰
 - Overview of users currently on-line
 - System documentation (items, codes, quality tests) automatically generated for the user
- Paper Coding Forms
 - Can be created both by Designer and by the individual user
 - Full control over fonts, colors, layout but still automatically created and always reflecting the current version of the Dictionary
 - Look like standard coding forms with boxes, options etc.
 - Can be produced filled with information in the current data table: individual reports can be printed as filled or partially filled paper coding forms; great for obtaining follow-up information or additional laboratory data
- Backups
 - Users can download a Microsoft Access backup of their own data at any time
 - The backup module contains full conversion to SPSS writing out the entire dictionary and creating the SPSS system files

⁸ Both Kaplan-Meier and simple COX models supported; this is not meant to be a replacement of a statistical package but a simple tool for quick data quality control

⁹ ProMISe automatically maintains a table with all modifications to all items during the life of a project. Hence it is known WHO modifies WHICH item from an OLD value into a NEW one on what DATE+TIME at which IP-address under WHICH USER name

¹⁰ The Center table, as part of the Dictionary, contains also addresses, telephone numbers etc. which are available to the normal ProMISe user also.

- Using the backup module, local private data management projects can be seamlessly integrated with the central, internet based project using standard Microsoft Access features (linked tables)

Properties of any ProMISe project from the viewpoint of the project Designer (super-user).

- Project Creation
 - Menu-based interface for the specification of
 - Project structure (relational structure)
 - Item properties
 - Label
 - Layout
 - Type
 - Coding system
 - Consistency checks
 - Coding System
 - Labels + codes
 - Synonyms
 - Extended labels
 - Chapter and Section divisions
 - Headers for subdivision of information
 - Logical Tests, Warnings and Jumps
 - Tests: values rejected based on logical conditions
 - Warnings: situations flagged based on logical conditions
 - Jumps: logical flow of data entry based on the contents of the data fields¹¹
 - Help files
 - GIF, DOC, HTML and ASCII supported
 - All info of the Dictionary maintained in transparent SQL-server tables
 - Dictionary can be made available to other applications in real time
 - Dictionary can be changed/extended at any time during the projects life time
 - All modifications to the Dictionary are fully automatically implemented in SQL Server by changing whatever is necessary in the data table definitions
- Project Maintenance
 - Web based interface for the maintenance of
 - Users
 - Centers
 - Registries
 - Help files
 - Bulk upload (bypassing manual data entry) fully supported
 - Intelligent upload creating records automatically
 - Maintains relational integrity of the project even if partial data are uploaded
 - Full reporting on changes
 - Partial changes possible
 - Designer may use a separate MS Access module to directly access all SQL-Server tables via ODBC¹²
 - Applications: VBA special-purpose applications developed by the Designer or his organization can directly change the data base tables also (pass-through queries)
 - Separate data tables for testing within the project
 - Separate Dictionary for testing within the project
 - LOG files provide information on
 - User activities (logon, execution of programs etc)
 - Data Modifications (old value, new value, date+time, user name, IP address)

¹¹ E.g. the skipping of non-applicable fields; even skipping over specific pages if not applicable in a certain situation

¹² Though not necessary, this may be a useful option: the Access MDB file is automatically generated by ProMISe and contains the link statements to all tables in a project. Hence the Designer can if necessary perform bulk operations on the data table which are not supported by the ProMISe user interface. Of course this can also be done using Query Analyzer or SQL-Server it self. However the Access module can access the SQL Server project over Internet using standard ODBC drives from anywhere in the world, provided the firewall of the Server opens the ODBC port to this particular user (IP-address)!

This document is not exhaustive. It tries to sketch the design and possibilities of ProMISe without going into technical details and without trying to deliver a complete overview. For specific and/or general questions about ProMISe and its possibilities in your specific situation, please contact me personally!

Leiden, January 2002

Ronald Brand
Senior Lecturer in Biostatistics
Dpt. of Medical Statistics
Section Advanced Data Management
Leiden University Medical Centre
Wassenaarseweg 62
2333 AL Leiden
P.O. Box 9604, 2300 RC Leiden, The Netherlands

Tel: (31)71 5276734
Secr.: (31)71 5276825
Fax: (31)71 5276799
E-mail Statistical issues: R.Brand@LUMC.NL
E-mail ProMISe issues: MRDM@ClinicalResearch.NL

ProMISe demo page: <http://www.clinicalresearch.nl/promise.htm>