MASTER THESIS

Master

Master in Environmental Engineering

Title

Efficient on-line monitoring of river water quality using automated measuring stations

Author

Queralt Plana Puig

Directors

Peter A. Vanrolleghem and Janelcy Alferes

Département de génie civil et de génie des eaux. Université Laval

Codirector

Jaume Puigagut

Universitat Politècnica de Catalunya

Date

February 2013

≪ Voici mon secret. Il est très simple:
on ne voit bien qu'avec le cœur.
L'essentiel est invisible pour les yeux.≫
Le Petit Prince, **Antoine de Saint-Exupéry (1943)**

Acknowledgements

En el moment que s'escriuen els agraïments, és un senyal de que s'arriba al final. Acabar aquesta tesi, és com posar punt i final a tota aquesta aventura. I només de pensar-hi, noto un buit. Haver pres la decisió de travessar l'oceà per a desenvolupar la tesi del màster a l'estranger ha fet que en poc temps m'hagi passat molta gent diferent per davant, a més d'haver-me fet créixer i descobrir món.

Potser semblarà estrany llegir uns agraïments multilingüístics, però m'agradaria agrair a totes aquelles persones que, d'una manera o una altra, han format part d'aquesta aventura en l'idioma corresponent.

Firstly, I want to thank Peter to give me the opportunity to work with his team. This experience made me grow up, both personally and professionally.

También tengo que agradecer a Janelcy, la postdoc con quien he estado trabajando, por guiar mi tesis hasta alcanzar sus hitos.

And obviously, I have to mention model*EAU* team, a big hard-working team. It was a pleasure to work with all of them.

Aussi, je ne peux pas oublier les gens qui ont fait que la vie à l'université était un peu heureuse. Merci à tous par tout, sans vous des petites pauses n'auraient pas été les mêmes.

També gràcies a en Jaume per acceptar la co-direcció de la tesi i donar-hi l'últim vist-i-plau.

Tampoc em puc oblidar de la Maria i en Lluís, al cap i a la fi ha sigut gràcies a ells que he tingut la oportunitat de treballar amb en Peter i amb tot l'equip de model*EAU*.

Molt especialment, també vull agrair a la meva família, als meus pares i al meu germà, que durant aquests mesos hi ha hagut un oceà entre mig però en cap moment m'ha mancat el seu recolzament.

I finalment, a tots els que també han posat el seu granet de sorra: Gràcies. Gracias. Thank you. Merci. Danke.

Abstract

Automated Measuring Stations for river quality monitoring have been used to generate on-line measurements of water quality variables. Even though a high degree of automation has been implemented at the levels of measurements, maintenance and control, data management and the resulting access to errors and uncertainties.

Besides managing a monitoring program, there is the responsibility to ensure high quality data with data and methods that are broadly accessible, and as cost-effective as possible. Moreover, higher assurance is needed to analyse, interpret, and apply data that are being collected so as to form monitoring networks that are designed to promote comparability of data across sites and across scales of space and time.

mon*EAU* (monitoring of water, "eau" in French) is the next generation of water quality monitoring networks. This project has as vision to provide some tools to get a flexible and open modular system, a high quality/performance database, remote use, automatic data quality assessment, user-friendly and user-oriented software, and proactive and flexible maintenance.

In the context of the project, the objective of this thesis was to efficiently monitor, collect and manipulate water quality data such that unreliable data due to anomalies in sensors, insufficient maintenance tasks, severe environmental conditions and other external factors can be minimized. To achieve these goals it was required to maintain and keep the monitoring stations working, collect the data, analyse samples in the laboratory to verify sensor data, create an appropriate database, import all collected data to a database, and finally export the data to statistics software and data manipulation programs for posterior analysis.

Resum

Estacions de Mesura Automàtica pel monitoreig de la qualitat del riu s'han utilitzat per a l'obtenció de mesures en línia de paràmetres de la qualitat de l'aigua. Encara que un alt grau d'automatització s'hagi implementat als nivells de mesura, manteniment i control, el tractament de dades i els resultats hereden errors i incerteses.

D'altra banda, manejar un programa de monitoreig inclou la responsabilitat d'assegurar dades qualitat, dades i mètodes àmpliament accessibles, i un programa el més rentable possible. A més, un alt compromís ès necessari per a analitzar, interpretar i aplicar les dades que han estat recol·lectades per xarxes de monitoreig que han estat dissenyades per promoure la comparació de dades de diversos llocs i de períodes diferents.

monEAU (monitoreig de l'aigua, "eau"en francès) forma part de la següent generació de xarxes de monitoreig de la qualitat de l'aigua. Aquest projecte té la visió de desenvolupar diverses eines per tal d'aconseguir un sistema modulat i flexible, una base de dades d'alta qualitat i d'alt funcionament, un ús remot, una avaluació de la qualitat de les dades automàtica, un concepte de software fàcil d'usar i orientat a l'usuari, i un manteniment proactiu i flexible.

En el context d'aquest projecte, l'objectiu principal de la tesi és monitorar, recollectar i manipular paràmetres de qualitat de l'aigua eficientment. Per tal d'aconseguir l'efectivitat cal reduir les dades poc fiables causades per anomalies als sensors, insuficients manteniments, condicions ambientals adverses i altres factors externs. Per aconseguir aquestes fites s'ha manteningut en bones condicions i en funcionament les estacions de monitoreig, col·lectat dades, analitzat mostres al laboratori per verificar les dades dels sensors, creat una base de dades apropiada, importat les dades recollides a la base de dades, i finalment exportat les dades a programes estadístics i de tractament de dades per a posteriors tractaments.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

AC	alternative current
ADQATs	automatic data quality assessment tools
AMS	automated measuring stations
APHA	American Public Health Association
API	application programming interface
API	application programming interfaces
ASTM	American Society for Testing and Materials
COD	chemical oxygen demand
DIN	Deutsches Institut für Normung
DMS	database management systems
EC	Environment Canada
EDSC	Environmental Data Standards Council
EPA	U.S. Environmental Protection Agency
EPA STORET	U.S. Environmental Protection Agency, Storage and Retrieval
GFCI	ground fault circuit interrupter
GUI	graphical user interface
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization of Standardization
LCL	lower control limit
LDO	luminiscent dissolved oxygen
LIMS	Laboratory Information Management System
LWL	lower warning limit
model*EAU*	Canada Research Chair on Water Quality Modelling

mon*EAU*	Automated Monitoring Station
NMKL	Nordisk Metodikkomité for Næringsmidle
NTC	negative temperature coefficient
NTU	nephelometric turbidity unit
ODBC	Object Database Connectivity
RAID	redundant array of independent disks
SOP	standard operation procedure
TOC	total organic carbon
TSS	total suspended solids
UCL	upper control limit
UMTS	universal mobile telecommunications system
USGS	U.S. Geological Survey
USGS NWIS	U.S. Geological Survey, National Water Information System
UV-VIS	ultraviolet-visible spectroscopy
UWL	upper warning limit
VBA	Visual Basic for Applications
WQ	water quality
WQM	water quality monitoring
WSDOT	Washington State Department of Transportation
WWTP	wastewater treatment plant

Chapter 1

Introduction

1.1 Need for environmental monitoring

The human activities that influence the environment have increased dramatically during the past few decades: terrestrial ecosystems, freshwater and marine environments and the atmosphere are all affected. The scale of socio-economic activities, urbanisation, industrial operations and agricultural production, has reached the point where, in addition to interfering with natural processes within the same watershed, they also have a world-wide impact on water resources. Consequently, a serious need has emerged for comprehensive and accurate assessments of trends in water quality, in order to raise awareness of the urgent need to address the consequences of present and futures threats of contamination and to provide a basis for action at all levels. Reliable monitoring data are indispensable basis for such assessments.

Environmental monitoring involves collecting and analysing physical, chemical, and/or biological information on the state of the environment in order, to identify changes and trends over time (EC, 2012; Corbitt, 1990). As an integral part of scientific research, it is also allows to verify whether policies and programs are having the desired results and activities are in compliance with legislation. There is a necessity of environmental monitoring as part of integrated environmental research programs, to design, implement, and evaluate effective environmental politics, to track natural resources.

However, there is no specific single best model for the structure of monitoring study. Rather, a successful program must be designed to survive lean periods by maintaining a solid funding base, a core set of inexpensive measurements, and a group of individuals dedicated to collecting, interpreting, and using data. Lovett et al. (2007) define beyond this personal or institutional commitment, successful monitoring programs have several important characteristics of design and implementation:

1. **Design the program around clear and compelling scientific questions.** These questions determine the variable measured, spatial extent of sampling, intensity and duration of the measurements, and the usefulness of the data.
2. **Include review, feedback, and adaptation in the design.** The guiding questions may change over time, and the measurements should be designed to accommodate such changes. The program should have the capability to be adapted

to changing questions and incorporate changing technology without losing the continuity of its core measurements.

3. **Choose measurements carefully and with the future in mind.** The measurements selected, the basic measures, or indicators of change, are important because not each variable can be monitored. Also if the question involves monitoring change in a statistical population, these measurements should be carefully chosen to provide a statistically representative sample of that population.

4. **Maintain quality and consistency of the data.** Sample collections and measurements should be rigorous, repeatable, well documented, and employ accepted methods. Methods should be changed only with great caution, and any changes should be recorded and accompanied by an extended period in which both the new and the old methods are used in parallel, to establish comparability.

5. **Plan for long-term data accessibility and sample archiving.** Metadata should provide all the relevant details of collection, analysis, and data reduction. Besides, raw data should be stored in an accessible form to allow new summaries or analyses if it is necessary. Raw data, metadata, and descriptions of procedures should be stored in multiple locations. Policies of confidentiality, data ownership, and data hold-back times should be established at the beginning.

6. **Continually examine, interpret, and present the monitoring data.** The best way to catch errors or notice trends is to use the data rigorously and often. Adequate resource should be committed to managing data and evaluating, interpreting, and publishing results. These are crucial components of successful monitoring programs, but planning for them often receives low priority compared to actual data collection.

7. **Include monitoring within an integrated research program.** An integrated program may include modelling, experimentation, and cross-site comparisons. This multi-faceted approach is the best way to ensure that the data are useful and, actually, are used.

Sometimes it is difficult to understand that monitoring is not a second-rate science. Rather, it is an essential component of environmental science and deserves the careful attention of scientists and from government agencies and other funding sources.

1.2 Water Quality Monitoring

Freshwater is a finite resource, fundamental for agriculture, industry and even human existence. Water quality (WQ) degradation is caused by discharge of toxic chemicals, overpumping of aquifers, long-range atmospheric transport of pollutants and contamination. Further, water pollution and wasteful use of freshwater threaten development projects and make water treatment essential in order to produce safe drinking water (Chapman, 1998). Therefore, water of good quality is crucial to sustainable socio-economic development. And it is for that reason that aquatic ecosystems are threatened on a world-wide scale by a variety of pollutants as well as destructive land-use or water-management practices. Efforts to improve or maintain a certain WQ often compromise between the quality and quantity demands of different users. There are increasing recognitions that natural ecosystems have a legitimate place in the consideration of options for WQ management.

Water quality monitoring (WQM) is the foundation on which WQ management is based and it is defined by the International Organization for Standardization (ISO) as: "the programmed process of sampling, measurement and subsequent recording or signalling or both, of various water characteristics, often with the aim of assessing conformity to specified objectives" (ISO, 2006). To determine these characteristics, water bodies can be fully characterised by three major components: hydrology, physico-chemistry, and biology. Furthermore, monitoring provides the information that permits rational decisions to be made on the following:

- Describing water resources and identifying actual and emerging problems of water pollution.
- Formulating plans and setting priorities for water quality management.
- Developing and implementing water quality management programmes.
- Evaluating the effectiveness of management actions.
- Building water supply networks.

According to different studies (Bartram and Ballance, 1996; Chapman, 1998), this general definition can be differentiated into three types of monitoring activities that distinguish between long-term, short-term and continuous monitoring programmes defined by the European Parliament (2000) as follows:

- *Monitoring* is the long-temp, standardised measurement and observation of the aquatic environment for a specific purpose.
- *Surveys* are finite duration, intensive programmes to measure and observe the quality of the aquatic environment for a specific purpose.
- *Surveillance* is continuous, specific measurement and observation for the purpose of water quality management and operational activities.

The continuous monitoring on-line/in-situ detection of pollutants in water and wastewater should be the best practice for the true quality monitoring (Thomas and Pouet, 2005). Monitoring, as a practical activity, provides the essential information which is required for an assessment of WQ. However, assessments require additional information, such as an understanding of the hydro-dynamics of a water body, information on geochemical, atmospheric and anthropogenic influences and the correct approaches for analysis and interpretation of the data generated during monitoring.

The concentration and state (dissolved and particulate) of some or all the organic and inorganic material present in the water is not only determined by in-situ measurements, it also requires an examination of water samples on site or in the laboratory. The main elements of WQM are, therefore, in-site measurements, the collection and analysis of grab samples, the study and evaluation of the analytical results, and the reporting of the findings. And one should remeember that the results of analyses performed on a single water sample are only valid for the particular location and time at which that sample was taken.

One purpose of a monitoring programme is, therefore, to collect sufficient data to evaluate spatial and/or temporal variations in WQ. Chapman (1998) suggests that a programme may need to be flexible to meet short-terms objectives but still be capable of developing over longer periods to meet new concerns and priorities. Besides, to achieve a successful programme to produce the expected information, Harmancioglu et al. (1998) proposes the basic steps of a data management system presented in figure 1.1.

In some studies, two basic functions are defined for WQM: prevention and abatement (Dandy and Moore, 1979; Karpuzcu et al., 1987). The first one has the objective of maintaining the existing unpolluted or acceptable status of WQ; while the second one puts the emphasis on a control mechanism by reducing or moderating pollution conditions. Prevention foresees the enforcement of effluent standards and, thereby, requires effluent monitoring plus trend monitoring. For abatement, compliance with in-stream standards is significant, so that compliance monitoring has the highest priority among other types of monitoring.

1.2.1 Objectives of water quality assessment

Before to start an assessment programme, a detailed inquiry about the real need for WQ information is required. Since water resources are usually subjected to several competing beneficial uses, monitoring which is used to acquire necessary information should reflect the data needs of the various users involved (Helmer, 1994). Accordingly, two different types of monitoring programmes exist, depending on how many assessment objectives have to be met:

- *Single-objective monitoring* which may be set up to address one problem area only. This involves a simple set of variables, such as: pH, alkalinity and some cations for acid rain; nutrients and chlorophyll pigments for eutrophication; various nitrogenous compounds for nitrate pollution; or sodium, calcium, chloride and a few other elements for irrigation.
- *Multi-objective monitoring* which may cover various water uses and provide data for more than one assessment programme, such as drinking water supply, industrial

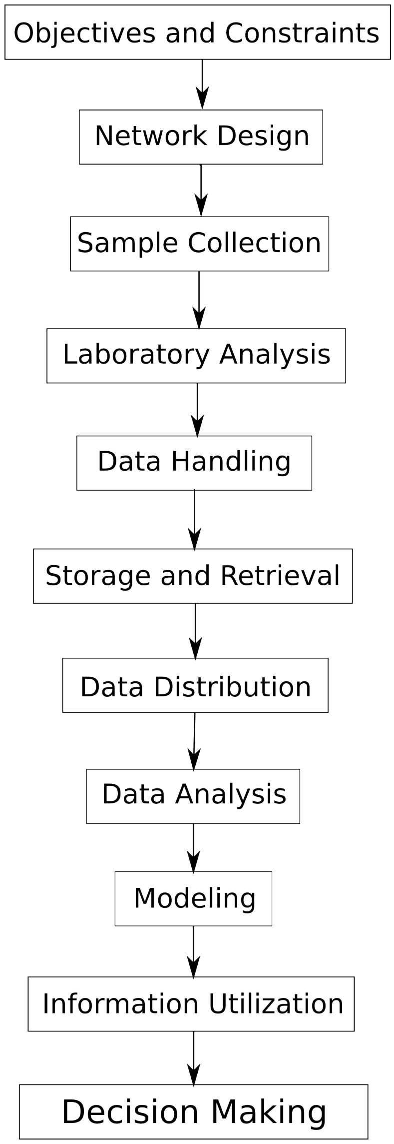


Figure 1.1: Basics steps in data management system (Harmancioglu et al., 1998)

manufacturing, modelling, control or decision making, thereby involving a large set of variables.

Chapman (1998) suggests that the implementation of the assessment programme objectives may focus on the spatial distribution of quality, on trends, or on pollutants. Introducing these three requirements is very costly and also, virtually, impossible. Subsequently, preliminary surveys are necessary to determine the necessary focus of any operational programme.

Once the objectives have been set, to determine the monitoring design, a review and interpretation of existing WQ data is required. This should be followed by recommendations to relevant water authorities for water management, water pollution control, and eventually the adjustment or modification of monitoring activities.

1.2.2 Complexity of water quality monitoring

Whatever the specific purpose of monitoring may be, first it must be recognized that WQM is a highly complex issue. Apart from technical features of monitoring, this complexity may be attributed to two factors:

1. Uncertainties in the nature of WQ.
2. Uncertainties in establishing a specific purpose of monitoring.

The nature uncertainties of WQ are a result of the two fundamental mechanisms underlying the hole processes: the natural hydrological cycle and man-made effects, which are often referred to as the *"impact of society"*.

Monitoring activities are required to reflect the stochastic nature of WQ to efficiently produce the expected information. These activities, considering the recent technological developments in analysis and sampling systems, can be carried out by Automated Measuring Stations (AMS) (Bols et al., 1999). Also, AMS generate a high resolution of datasets. For that reason, most researchers like Sanders (1983); Karpuzcu et al. (1987) specify the term *monitoring* further to mean *statistical sampling*.

Comparing other monitoring projects (e.g. Beck et al. (1998); Vandenberghe et al. (2005)), similar problems are observed. Unfortunately, most of the failure information is not available for everyone, as it is only within reach of personal communication or it has to be deduced from "between the lines" of any publications. Rieger and Vanrolleghem (2008) conclude that the three major reasons limiting the use of monitoring stations are:

1. The lack of standardization.
2. Data quality problems, which lead to data graveyards that do not provide the required or useful information
3. Insufficient flexibility of the stations leading to problems when new or better sensors should be connected or when the focus of the project changes.

As complex as it is, WQM is also highly significant because it is the only means of being informed about WQ. Thus, monitoring constitutes the link between the actual process and the understanding, interpretation, and assessment of the highly complex phenomena. Therefore, WQM is the most crucial activity on man's side with respect to all management and control efforts. Adequate and reliable data may serve to increase the knowledge on environmental processes and hence reduce the uncertainties whereas lack of such data may lead to erroneous interpretations and decisions (Harmancioglu and Alpaslan, 1992).

1.2.3 Need for Quality Data

With the increasing use of on-line sensors for WQ measurements, a change can be observed from not having enough data to have plenty or even too much data. Whereas the accuracy of the lab measurements is normally sufficient, urgently needed data quality evaluation concepts for the continuous measuring devices are not available or inefficient in day-to-day operation (Rieger and Vanrolleghem, 2008).

Monitoring data have been accumulated in file cabinets (or on hard drives) and gathered dust. These data may be either inaccessible to all but a few, too poor in quality or too poorly documented to be useful, or they may have been collected exclusively to fulfil a legal requirement, with no real motivation for thorough analysis and interpretation (Lovett et al., 2007). At the other end of the spectrum are datasets from both individual investigators and large institutional programs that have enormous value to environmental science and policy.

The importance of the use of information should be stressed. It is essential that the design, structure, implementation and interpretation of monitoring systems and data are conducted with reference to the final use of the informations for specific purposes. Thus, information on WQ processes is needed with respect to water resources management in general and to pollution control in particular. Retrieval of such information requires collection of data; basically, the purpose of data collection practices is to produce the information needed for efficient management of the water environment (Harmancioglu, 1997).

In the past, monitoring activities were carried out in a problem, project, or user-oriented framework. Recently, as the emphasis is shifted more to WQ management and control efforts in a larger perspective, Harmancioglu et al. (1999) assured that the major concern has become the assessment of the quality of surface waters in an extensive area or a river basin. In achieving this specific purpose, trend monitoring is required to evaluate both the changing quality conditions and the results of control measures.

Networks for WQM must conform programme objectives. A clear statement of objectives is necessary to ensure collection of all necessary data and to avoid needless and wasteful expenditure of time, effort and money. Furthermore, evaluation of the data collected will provide a basis for judging the extent to which programme objectives were achieved and thus justify the undertaking. Before observations begin, it is also essential to specify the locations of sampling stations, the frequency of sampling and WQ variables to be determined. According EPA (2012a), European Parliament (2000) and

Chapman (1998), the most common measured parameters to check the WQ are:

- Hydrology
  - Flow level
  - Velocity
  - Flow
- Physico-chemistry
  - Temperature
  - Conductivity
  - pH
  - Dissolved oxygen (DO)
  - Alkalinity
  - Total hardness
  - Total suspended solids (TSS)
  - Turbidity
  - Total organic carbon (TOC)
  - Chemical oxygen demand (COD)
  - Ammonia-nitrogen
  - Nitrate+nitrite-nitrogen
  - Total phosphorus

To fulfil these functions, it is necessary to do some preliminary data analyses to provide basic background knowledge of existing WQ conditions. After gathering sufficient data, Chapman (1998) mentioned that it is possible to describe the average conditions, the variations from average and the extremes of WQ, expressed in terms of measurable physical, chemical and biological variables. Not forgetting at the same time to set priorities, make plans and implement management programs. Ultimately, monitoring programmes should be periodically reviewed to ensure that information needs are being met. As greater knowledge of conditions in the aquatic system is gained, a need for additional information may become apparent. Additionally, reviewing monitoring programmes avoids collecting unnecessary information.

The collection of data should be uniform to ensure compatibility and allow to apply the experience gained in any location. If networks for WQM are developed in close co-operation with other agencies actively collecting water data, this will not only minimise the cost of establishing and operating the network but also facilitate the interpretation of WQ data.

Moreover, it is necessary to emphasize that managing a monitoring program requires ensuring that the data are of high quality, the data and the methods are broadly accessible, and the program is as cost-effective as possible. Also it requires the responsibility to make the commitment to maintain and expand long-term monitoring programs. Consequently, this maintenance of long-term monitoring datasets should be a highly valued feature in the review of proposals.

Concluding, to collect and maintain high quality data a robust system is required guaranteeing atomicity, consistency, isolation and durability. A good system that offers these properties to satisfy the necessity of quality data is a database. International organizations like EPA (2012b) or USGS (2002) are using this tool to manage and treat large amounts of WQ data. They also provide some tools to import and export the data to create good quality of data series to optimize the data evaluation process.

1.3 monEAU project

Nowadays, the on-line sensors for WQM of different water bodies are improving and its use is increasing to reach different goals. Compared to grab or composite samples, automatic data acquisition systems can afford high measuring frequencies allowing a better description of the dynamics in reactive water bodies (river, combined sewer, wastewater treatment plants (WWTP), etc.). Consequently, a huge amount of data can be produced, but they are of uncertain quality. Unfortunately manual data validation requires time and it is very dreary. As a result, automatic data quality assessment tools (ADQATs) are necessary to validate time series and to use them for their meant application. In that sense, poor quality data could drastically affect the results of their application, namely water quality models, WWTP control rules, etc.

mon*EAU* (monitoring of water, "eau" in French) is the next generation of water quality multi-objective monitoring networks (van Griensven et al., 2000). In this project, ADQATs are being developed and applied to a real case to evaluate their performance in terms of producing good experimental data quality. They are applied to an in-situ monitoring stations measuring the water quality dynamics during rain events affecting a small urban river.

As Rieger and Vanrolleghem (2008) describe, this project pursues a vision to develop some elements to design new products. The basic features are:

- *A Flexible System.* To use the station for different and multiple monitoring and research goals, at different locations (e.g. river, WWTP, sewer or use for collection of meteo and hydraulics data), with a wide array of sensors and sampling methods (in-situ/on-line/off-line), and with all standard communication protocols for sensor connections.
- *An Open and Modular System.* To adapt the system to special demands and to

add special features and at the same time guarantee the integrity of the framework. This created framework is a robust software with basic functionality.

- *A High Quality/Performance Database.* To store the large data series, the database structure needs to provide sufficiently fast access but be flexible enough for any monitoring task and further developments of the station's functionality.
- *Remote Use.* Not always the monitoring stations are easy accessible for the users. The design has to consider the minimum maintenance requirements as possible, minimization of energy demand in combination with different power options, various telemetry options, remote access to sensors, and remote access to monitoring station operation.
- *Automatic Data Quality Assessment.* Reference samples, sensor status/diagnosis data and time series information providing redundant data over time, space and determinants.
- *User-Friendly and User-Oriented Software Concept.* The required information is provided and visualized depending on the user level and the location. For instance, the operator at the base station needs information criteria for maintenance and recalibration. In that way, the expert working at the central server should get access to all information of all base stations connected to it, permitting decisions on remote calibration, changes in operational setting or relocation of a base station.
- *Proactive and Flexible Maintenance Concept.* Information on maintenance and a schedule for the operators to execute it based on the sensor self-diagnosis, the company or user experience and a proactive set of station-triggered experiments.

Chapter 2

Objective

In the context of mon*EAU* project, the aim of this thesis is to efficiently monitor, collect and manipulate WQ parameters measured by AMS.

Minimizing unreliable data caused by anomalies in sensors, insufficient maintenance tasks, severe environmental conditions and other external factors is the main object to reach its effectiveness.

Specifically, to achieve these goals several activities are developed:

- Maintain and keep the monitoring stations working.
- Collect WQ data.
- Analyse samples in the laboratory to validate sensor data.
- Create an appropriate database.
- Import collected data to the database.
- Export the data to statistics and manipulation data programs.

Chapter 3

Materials and Methods

This section will describe the materials that make up the stations to recollect the WQ data, as well as all methods used to develop this thesis and achieve its goals.

3.1 Site description

Scientists and environmentalists do not monitor rivers to find out if they are polluted, but rather to find out how polluted they are, and which is the pollutant.

Since the mon*EAU* system can be implemented wherever it is desired and because rivers have a special interest, in the current implementation, the mon*EAU* stations have been installed at the small Notre-Dame river situated in Quebec City next to the Jean Lesage International Airport (Coordinates: 46°48'19,81"N, 71°22'04,95"O) with the objective of observing the impact of the housing and small companies on the hydraulics and pollution profiles.

Two automatic monitoring stations have been installed separated by 2 km, the downstream station and the upstream station (Figure 3.1). Each station has two separate parts: the equipment to measure the WQ parameters and the instruments to measure the hydrology situated 50 m downstream f the station.

3.2 Station description

5 According the mon*EAU* vision, the monitoring stations should be a versatile design of water side monitoring equipment. It gives flexibility to be deployed wherever it is wanted and provides a software framework and an open-code structure that will allow advanced users the ability to customize the output that is generated.

To measure the WQ in real-time, the monitoring stations RSM30 built by Primodal Systems were used. The RSM30 is a state-of-art WQM station that registers, transmits and analyses the data in real-time using the custom-designed software inside (Primodal, 2012).

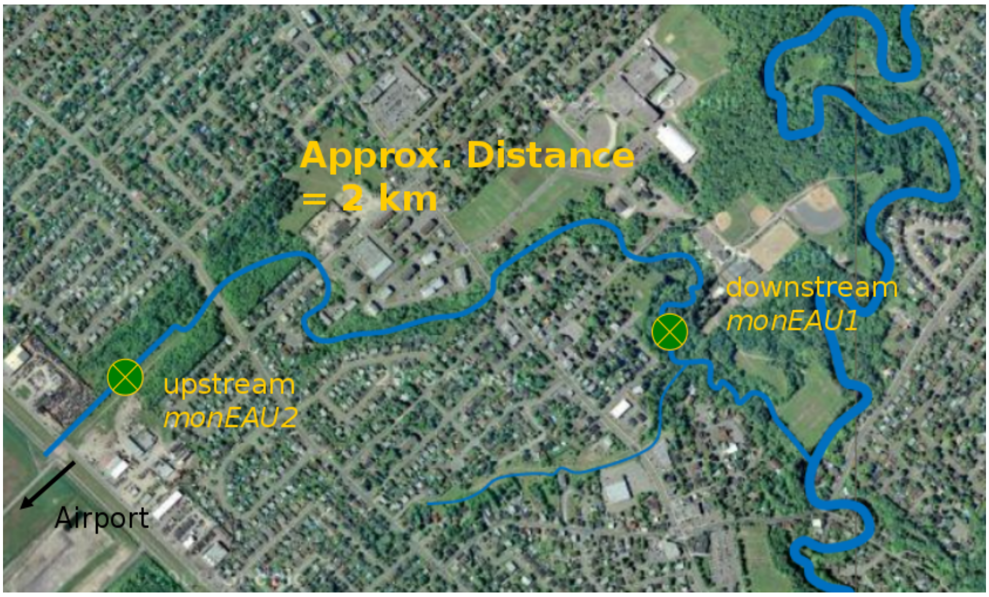


Figure 3.1: Site where the monEAU stations are installed.

The RSM30 is encased in a secure NEMA 4X rated fibreglass enclosure designed for a range of diverse environmental conditions and protected from the vandalism by a cage (Figure 3.2).

Suitable for year-round outdoor use, the station has been designed for rapid and ease-of-operation deployment, it is equipped with handles and castors to simplify its movement and portability. Also, climate control options can be included to mitigate extreme temperature fluctuations, several power source options such as a simple 110V or 240V plug, battery or solar panel. Moreover, equipment safety is assured through surge and GFCI (Ground Fault Circuit Interrupter) protection inside the unit.

The RSM30 offers different communication options to control the communication with the system. One is directly interfacing with the unit using the RSM30's internal components when a technician is on-site. Alternatively, the communication can be achieved using the wireless capability built into the system. That permits the unit to be connected to an existing local wireless network from the laptop. Moreover, for communication from a remote location and for transfer of measurement data back to the Central Server it is possible to use its GSM (Global System for Mobile Communications) capability.

To measure the flow rate in real-time, a ultrasonic ow measuring instrument by Sigma (Hach, 2012b) is used. To protect the flow meter against rain and vandalism, it is placed inside a box that is locked and fixed.

Figure 3.2: Installed RSM30 in protective cage.

3.2.1 Software

Behind the mon*EAU* system, as Rieger and Vanrolleghem (2008) mentioned, there is a robust software framework serving as the mainstay of the stations and server network permitting the simple connection of various modules through a specified API (Application Programming Interface). Some modules will provide basic functionality like data input or output but the main reason for this framework structure is the ability to integrate new developments or to connect third party modules. In this way, robust operation (the framework is not open to the end-users) is combined with the required flexibility. Figure 3.3 shows the mon*EAU* concept.

In case of these stations, measurement, meta, LIMS (Laboratory Information Management System) and log data from sensors are controlled by Primodal's own PrecisionNow BaseStation software and ana::pro software (Advanced Process Software) provided by s::can (S::can, 2012). In case of the flow meter, the software used is called InSight (Hach, 2012a).

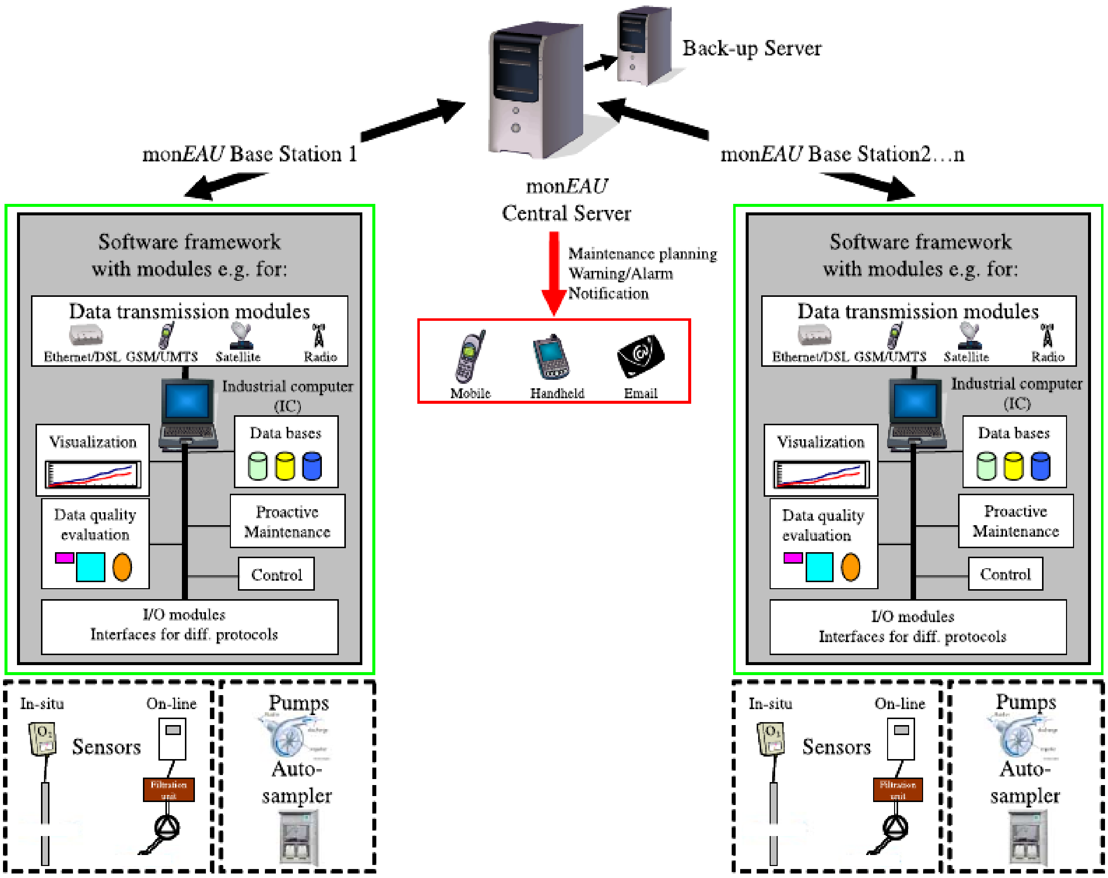


Figure 3.3: Set-up of the mon*EAU* WQM network.

PrecisionNow software

The PrecisionNow graphical user interface (GUI) provided by Primodal (Hamilton, Canada) is used to configure sensor inputs and data evaluation modules, along with setting up communication and data visualization (Figure 3.4). Current as well as historical time series data can be visualized inside the software as single or as multiple time series on a single graph. Or alternatively, the data can be exported in a variety of file formats including text, xml, csv or Excel files (Primodal, 2012).

The station can operate alone or as part of a monitoring network. Each unit comes fully equipped with all the required components to operate in isolation including the ability to store and visualise measurement and meta data, as well as the communication tools to transmit the data to a central storage location. Moreover, the RSM30 has sufficient capacity to store years of data.

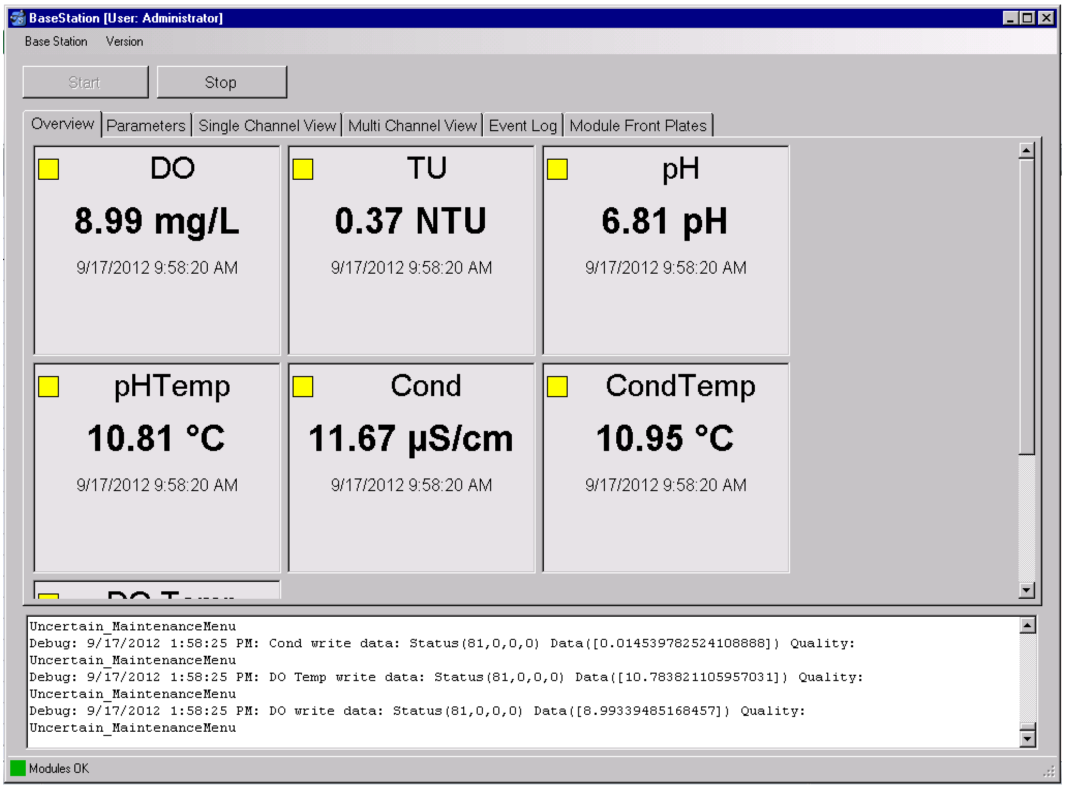


Figure 3.4: Basestation software interface

The structure of the RSM system includes the provision for the transmission to and storage of data on a Central Server. Like the Base Stations at the water's edge, the Central Server is equipped with a full complement of features including the ability to store and visualise measurements and meta data. Maintenance scheduling can be programmed and data evaluation modules can be used to analyse the raw data for any number of purposes such as identifying sensor problems, signalling alarms, or preparing the data for long-term storage.

The Central Server is programmed to pull data from any all Base Stations within the network at the time interval chosen by the user. This functionality enables the automated comparison of the data from multiple locations in real-time. Analysing the data from multiple locations in real-time will minimise the effort needed for post-processing and the manual comparison of data.

ana::pro software

The ana::pro software is created by s::can (Vienna, Austria). This advanced GUI offers numerical and graphical data and advanced multiparameter process visualization

(Figure 3.5). Likewise, it offers advanced spectral analysis, derivative and delta spectra access, an autocalibration module, data-logger access, automatic or manual transfer data, interpretation of measurement, off-line data analysis, interfaces for data transfer and automatic verification (S::can, 2006).

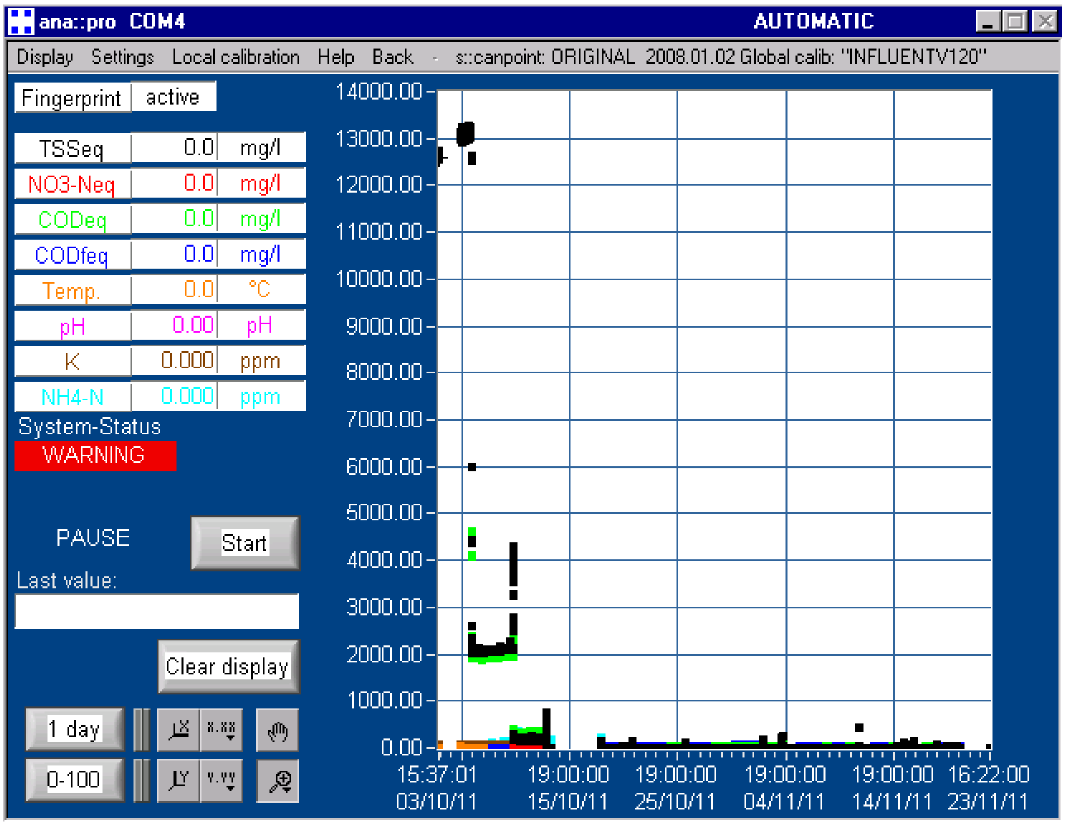


Figure 3.5: ana::pro software interface

It is important to emphasize that the ana::pro software is specially developed for the operation of all s::can spectrometer probes in more complex applications. In addition, it allows the operation of the s::can dissolved oxygen probes, the s::can ammonium probe and other sensors distributed by s::can.

3.2.2 InSight software

Finally, the InSight software for the Sigma flow meter is provided by American Sigma (Loveland, United States). This software provides easy-to-use, automated data collection and a management for multiple applications (Figure 3.6). It can communicate, download and analyse data from Sigma Flow Monitoring Systems. Additionally, it supports data retrieval, real time viewing of the logger status, multiple sensor support, program templates, remote programming, modem scheduling, and alarms (Hach, 2012a).

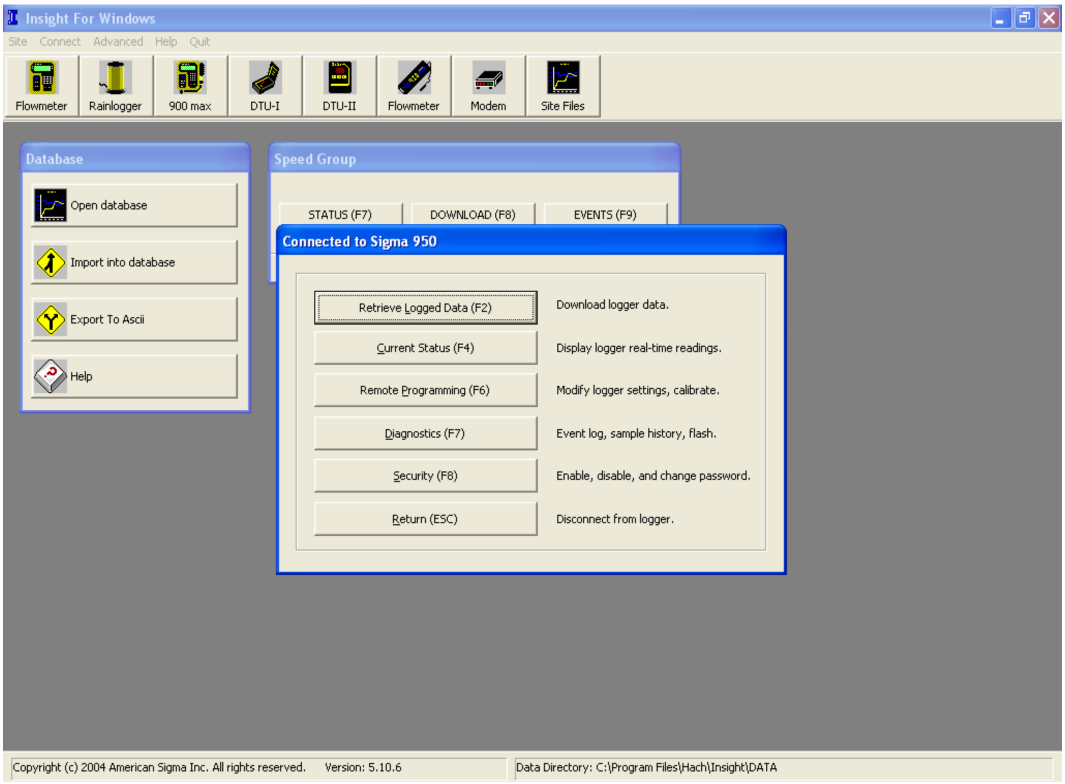


Figure 3.6: InSight software interface

On InSight, program templates store all programming steps in a file on disk, allowing re-use of common programs quickly and easily. To bring the logged data into the InSight software by retrieving it from a Sigma data logger can be done via modem or direct RS232 connection.

3.2.3 Sensors

A short brief about the sensors used in this implementation is presented, even if the mon*EAU* system is flexible for most of types of sensors.

Each installed station included six sensors to measure WQ parameters in example pH, conductivity, turbidity, ammonia, TSS, DO, temperature, nitrate, TOC and COD. To protect the sensors, they were caged in a probe housing as it is shown in the figure 3.7. Independently of the stations, another sensor was installed to evaluate the hydrology of the river in each point (i.e. flow level, velocity and flow).

Figure 3.7: Installed sensors in the protective cage.

pH sensor

The pH differential sensor used is manufactured by Hach. It has an integral NTC (Negative Temperature Coefficient) 300 ohm thermistor to automatically compensate pH readings for temperature changes.

The operating principle of the pH sensor described in its manual (Hach, 2006b) is to measure the pH value as an electrical potential in mV between the glass electrode and the reference electrode, similar to the potential between the two plates of a capacitor. The glass electrode acts as a transducer that transforms a chemical energy into electrical energy producing a potential proportional to the pH value.

The manufacturer offers one and two point automatic or manual calibrations for the pH sensor. The used calibration in this installation has been the two point manual one and it is performed by placing the pH sensor in two different buffers solutions with a known pH values (7 and 10). The procedure is described in modelEAU (2012d).

Conductivity sensor

The conductivity sensor installed is manufactured by Hach. The measurement of the inductive conductivity is made by passing an AC current through a toroidal drive coil which induces a current in the electrolyte solution. This induced solution current produces a current in a second toroidal coil. The amount of current induced in the second coil is proportional to the solution conductivity (Hach, 2008a).

There are three different options to calibrate the conductivity sensor: sample cal method, conductivity cal method and zero cal method. The used calibration in this case has been the sample cal method, placing the sensor in a solution with a known conductivity value determined by laboratory analyses. The procedure used is detailed in modelEAU (2012a).

Dissolved oxygen sensor

The dissolved oxygen sensor is a Luminescent Dissolved Oxygen (LDO) sensor and it is produced by Hach. This sensor is specially designed for municipal and industrial wastewater applications (Hach, 2006a).

Blue light from an LED is transmitted to the sensor surface. The blue light excites the luminescent material. As the material relaxes it emits red light. The time for the red light to be emitted is measured. Between the flashes of blue light, a red LED is flashed on the sensor and used as an internal reference. Increased oxygen in the sample decreases the time it takes for the red light to be emitted. The time measurements correlate to the oxygen concentration.

The LDO sensor has four different options for calibration: calibration in air, calibration by comparison to a Winkler Titration, calibration by comparison to a hand-held DO analyser and concurrent calibration of two sensors. In the present case, the calibration in air is used because it is the simplest and the most accurate method of calibration. More details are in modelEAU (2012c).

Turbidity sensor

The Solitax sensor provided by Hach can measure the two correlated parameters turbidity and TSS.

The measuring principle is based on a combined infrared absorption scattered light technique that measures the lowest turbidity values just as precisely and continuously as high sludge content (Hach, 2009). Using this method, the light scattered sideways by the turbidity particles is measured over an angle of 90°.

There are two calibration techniques; depending on whether turbidity or suspended solid is required. Calibration for turbidity requires the use of Turbidity Standard Solutions to develop calibration curves (Bertrand-Krajewski et al., 2007). The procedure step by step is indicated in modelEAU (2012e).

On the other hand, calibration for suspended solids requires calibration to the actual sample. This optimizes the compensation for the particle size and typical shape at the measuring site. It is best performed by keeping the sensor as usual for normal measurement, and then use grab samples that are evaluated by laboratory methods.

Spectro::lyser sensor

The spectro::lyser sensor manufactured by s::can allows measuring different parameters. In this case the measured parameters are nitrate, Total Organic Carbon (TOC), Chemical Oxygen Demand (COD) and Total Suspended Solids (TSS) or turbidity.

The spectrometer probes work according to the principle of UV-VIS spectrometry (S::can, 2007b). Substances contained in the medium to be measured weaken a light beam that moves through this medium. The light beam is emitted by a lamp, and after contact with the medium its intensity is measured by a detector over a range of wavelengths (220-720 nm or 220-390 nm). Each molecule of a dissolved substance absorbs radiation at a certain and known wavelength. The concentration of substances contained determines the level of absorption of the sample.

The length of the optical measuring path of the s::can spectrometer probe is fixed and cannot be varied. However, in case of a measuring path of 35 or 100 nm it can be reduced by inserting a shortening path length device.

To calibrate the spectro::lyser a local calibration is performed. The local calibration consists of presenting the specific composition of the measuring medium to the sensor with the help of comparative readings from laboratory analysis (modelEAU, 2012f). The composition of the measuring medium has to be determined from a sample prior to local calibration.

The global calibration is provided by the company and it contains the algorithms for the calculation of the concentrations from the spectrometric fingerprints. These algorithms are developed on the basis of s::can experiences with many comparable and representative measuring media. Additionally, for special measuring tasks or higher accuracy the local calibrations should be carried out.

Ammo::lyser sensor

The ammo::lyser sensor is manufactured by S::can. It monitors the concentration of ammonium and potassium ions in-situ using an ammonium and potassium selective electrodes. A robust ion selective membrane in the ammonium electrode separates the ammonium ions from the water. To compensate automatically for possible interferences, the ammo::lyser is equipped with sensors for pH and temperature and potassium electrode. The readings of these sensors can also be displayed online (S::can, 2007a).

The ion selective sensor uses a membrane that is porous for one specific ion-type. The combination of this selective membrane with the electrolyte inside the electrode allows measuring the redox potential corresponding with one specific ion. After the voltage measurement, the ammo::lyser calculates the concentration itself using the Nernst equation.

The calibration of the ammo::lyser is more laborious in comparison with the other sensors. Each parameter has to be calibrated individually. Because of the interdependence between the parameters it is recommended to calibrate them in the following order:

- Temperature (T)
- pH
- Potassium (K^+)
- Ammonium (NH_4^+)

Also, when only a single parameter requires recalibration, it is not necessary to recalibrate the other parameters afterwards. Only if the reference electrode is replaced all parameters, except for the temperature, have to be recalibrated. When more than one parameter is recalibrated, they should be recalibrated in the order presented above.

For all parameters an off-set (one point) calibration or a two-point calibration can be performed.

The calibration of the individual parameters on the s::can ammo::lyser can be performed using the advanced mode of ana::pro. The adjustment of the temperature calibration is best performed in-situ against a reference thermometer. In case of pH calibration, it is performed by placing the pH sensor in solutions with known pH. And finally, for the calibration of the ion selective sensors, it is strongly recommended to use the real values on site obtained by taking samples and determinate the two ions in the lab (details in modelEAU (2012b)).

Moreover, before starting a calibration, the following issues should be verified:

- All protective coverings are removed from the electrodes.
- No air bubbles are present in the round glass end of the pH electrode.
- To ensure that no air bubbles are present inside the membrane area of the ion selective electrodes, keep the ammo::lyser upright with one hand and with the other, tap sidewards several times against the basket guard of the electrodes.
- The membranes of the ion selective electrode should not be completely dry.
- For practical purposes it is best to submerge the entire sensor head in the calibration medium, even when only a single electrode is calibrated.

Flow Meter

To study the hydrology of the river, the Sigma Flow Meter is used. It is a portable, completely self-contained and sealed system.

The flow meter measures the liquid level in a channel that is directly contributing to flow and calculates the flow rate based on the head-to-flow relationship of the primary device (Hach, 2008b). Also it can measure the average velocity of the flow stream using a submerged Doppler sensor and calculates flow based on the current level and the following equation:

$$Wetted\ Area \times Velocity = Flow \tag{3.1}$$

The Doppler sensor measures the frequency shifts caused by the liquid flow. Two transducers are mounted in a case attached to one side of the pipe. A signal of known frequency is sent to the liquid to be measured. Solids, bubbles, or any discontinuity in the liquid, cause the pulse to be reflected to the receiver element and then the frequency of the returned pulse is shifted. The frequency shift is proportional to the liquid's velocity (modelEAU, 2011b).

To measure the water level a bubbler sensor is used. A small plastic tube is placed at the bottom of the open channel. Pressure variations in the tubing are proportional to the liquid level in the flow stream.

In the downstream monEAU station, the level, the average velocity and the flow are measured. However, in the upstream monEAU station, the measured parameter is only the level. It is not possible to measure the average velocity and the flow only with the bubbler sensor because the perpendicular section is not regular as in mon$EAU1$. Then, the flow is calculated by the velocity measured by a micropropeller and the geometrical calculus of the perpendicular section of the river determined by bathymetry.

3.2.4 Sensor controllers

The Hach sc1000 Multi-parameter Universal Controller is used. It is a state-of-the-art controller system with the possibility to use it directly with 8 sensors or network several together to accommodate many more sensors and parameters.

The controller consists of a display module and one or more probe modules. The display modules is an interface and large colour touch-screen display and it can be used for any number of parameters. In normal operation the touch screen displays the measured values for the selected probes (Figure 3.8). One display module can control one or several probe modules connected by a digital network. Each probe module can be configured with relays, analog outputs, analog or digital inputs, and digital fieldbus cards.

The display offers different display modes and a pop-up toolbar:

Figure 3.8: Interface and colour touch-screen display of controller sc1000.

- **Measured value display**: The controller identifies the connected probes and displays the associated measurements.
- **Graph display**: Displays measured values as graphs.
- **Main menu display**: Software interface for setting up parameters and settings of a device, probe and display module (Figure 3.9).
- **Pop-up toolbar**: The pop-up toolbar provides access to the sc1000 controller and probe settings and is normally hidden from view.

3.2.5 Data transmission protocols

With respect to the data transmission, Rieger and Vanrolleghem (2008) proposes a protocol as follows:

Sensor ⟷ Base Station: Even though divers communication protocols are of interest in WQ sensor systems, to make use of all available information coming from the sensor itself requires a bus protocol.

Base Station ⟷ Central Server: To connect the base stations with the central server, there are different telecommunication interfaces. For example telephone line, xDSL, cable TV, GSM/UMTS, dedicated radio link and satellite.

Database Structure and Safety for all meta data: To guarantee the readability of the data by different software and also to allow data use in the future, the IEEE 754 (Institute of Electrical and Electronics Engineers Standard for Floating-Point Arithmetic)

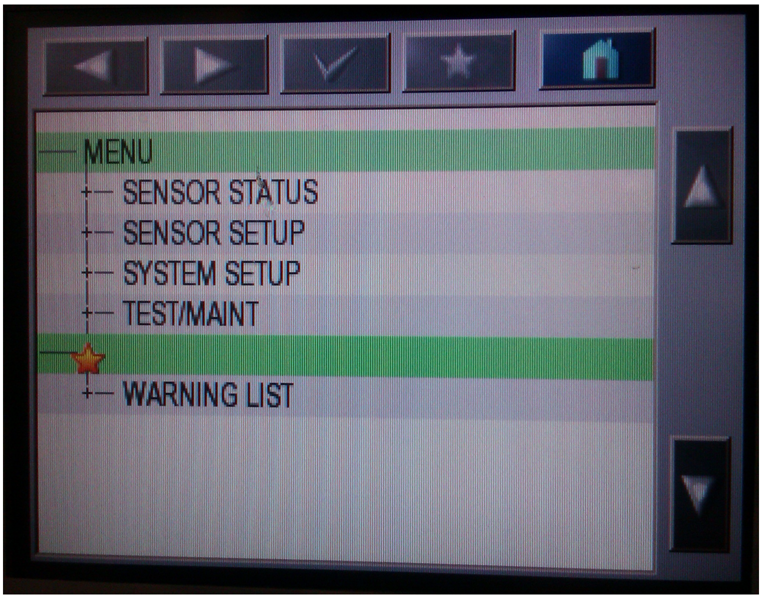


Figure 3.9: Main menu display of controller sc1000.

floating point standard format has been chosen. Moreover, to permit easy exchange of data with other software platforms different export functions are available.

Guarantee Data Safety: Some safety measures are integrated in the base stations as well as on the central server. In these measures, RAID (redundant array of independent disks) and sufficient hard disk space at the base stations are included to bridge communication breakdowns or store measurement data.

Data Quality Evaluation: When developing a data quality evaluation concept, a larger problem is that it should use as much information as possible while still being flexible for use at different locations. The mon*EAU* system includes a generic concept but encompasses the possibility of further integration of knowledge-based approaches.

3.3 Maintenance and operation

Every analytical method is composed of procedural, measuring, calibrating and evaluating instructions as the ISO (1990) defines.

The station requires periodical control and maintenance visits. A functional check might be required for one of the following reasons:

- Routine functional check.
- Suspicion of fouling of the measuring windows and electrodes.

- Change of location where probe is deployed or in type of probe.
- Suspicion of probe malfunction.
- Rain event.
- Changes in operational conditions.

During the execution of a functional check, the following actions have to be performed. The frequency depends mainly on the type of application.

- Checking actual status and the functionality of the probe.
- Checking the credibility of readings.
- Checking historical status or system stability.
- Checking unintentional modifications of measuring settings caused by unauthorised access or remote control.
- Checking the probe's mounting.

To carry out the functional check activities, Standard Operation Procedures (SOPs), a cleaning protocol, a calibration protocol, and control charts are created. Additionally, the maintenance and control procedure permits to assure the reliability of the measured values.

3.3.1 Standard operation procedure

For each sensor and software a SOP has been written by model*EAU* group with all relevant steps to execute maintenance, cleaning and calibration activities without the necessity to use the manual every time that one of these activities is required.

Each SOP includes:

1. *Introduction:* Summary about the name of the sensor, the measured parameters and the manufacturer.
2. *Definition and principle:* Description about the sensor and measured parameter/s, the principle of the operation of the sensor and the most common ranges of each parameter.
3. *Maintenance:* Specific properties of the sensor to keep it safe and instructions on how often clean and calibration are necessary.
4. *Calibration:* All steps required to calibrate the sensor.
5. *Cleaning:* All steps required to clean the sensor and the adequate material needed.

3.3.2 Cleaning protocol

Automated maintenance is limited to the self-cleaning using compressed air. With this respect, one of the main problems with the AMS is related to the adhesion of silt and clay on the sensors.

Each manufacturer defines the frequency of the cleaning activity for every sensor. However, depending on the application, the frequency can be different to that suggested by the manufacturer. In this implementation, as the application is in a river and the filth can be higher, the frequency of manual cleaning for all sensors has been established as twice per week. On the other hand, not only the frequency depends on the application, but also depending on the weather conditions (rain events).

Before cleaning the sensors, remove any large material or sediments that have been attached to the probe housing is to be removed. Afterwards, the cleaning can be done using a soft, wet cloth (Kim Wipe) and distilled water. Particularly, the lens of optical sensors have to be cleaned with a weak acid when they are noticeably dirty. The procedure is detailed in modelEAU (2011a).

3.3.3 Calibration protocol

Regular calibrations are required due to sensor and membrane damage, deterioration and misconfiguration. To verify whether calibration is necessary, the displayed readings should be compared with the values of a reliable comparison method (Section 3.3.4). In case of a significant difference between the laboratory values and the readings of the sensor, a calibration has to be performed.

pH, conductivity and LDO sensors are calibrated through the sc1000 controller. On the other hand, the spectro::lyser and the ammo::lyser are calibrated through the ana::pro software.

As cleaning, the sensor manufacturers recommend a frequency of the calibration activity. Nevertheless, in the use of application in the river, it has been necessary to calibrate more often the sensors due to the aggressiveness of the water. Additionally, the periodicity depends on the weather conditions.

As for cleaning, the sensor manufacturers recommend a frequency of calibration. Nevertheless, in the river application, it has been necessary to calibrate the sensors more often due to the aggressiveness of the water. Additionally, the frequency depends on the weather conditions.

For the more sensitive sensors, like pH, conductivity, spectro::lyser and ammo::lyser, a calibration is required every one or two weeks. For the LDO sensor, the frequency is lower, the need for calibration is every two or three weeks. On the other hand, the

Solitax doesn't require any calibration during the operation period since it is calibrated by the company.

3.3.4 Control charts

One of the main challenges for AMS is to operate according to standard guidelines. When different error sources can act simultaneously on the output, the validation and calibration of a station is a complex problem. The main goal of the concept is an in-control measuring process, which has to prove that the measurements are within a certain uncertainty range.

To validate a station requires a quality control procedure. Commonly, control charts are used to check data quality. Montgomery (2008) defines control charts as an on-line process-monitoring technique widely used to detect the occurrence of assignable causes of process shifts and estimate how the system is working. This general theory of control charts was first proposed by Walter A. Shewhart, and control charts developed according to these principles are called Shewhart control charts.

A general model for a control chart is the following. As the Figure 3.10 shows, typically, there are three control lines: a central line, an upper control limit (UCL), and a lower control limit (LCL). The central line is a measure of the general level of the process, and the UCL and LCL are used to help judge whether the process is operating in a state of statistical control.

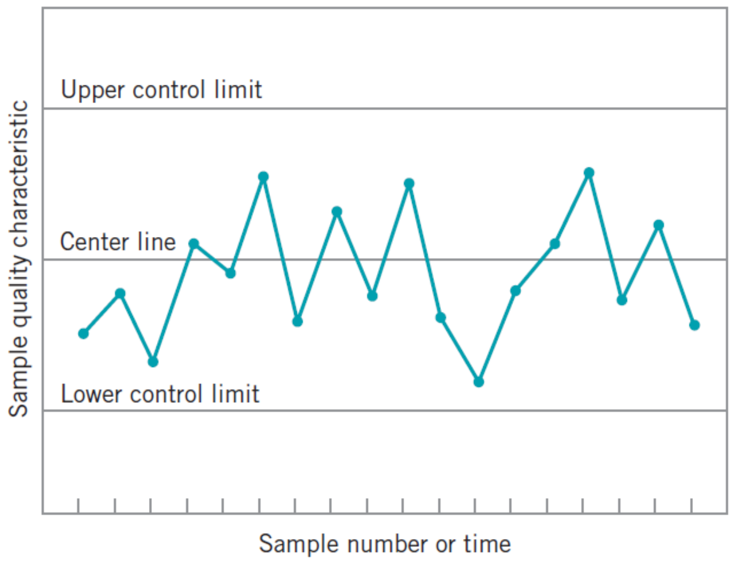


Figure 3.10: Model of a typical control chart (Montgomery, 2008).

Assuming that a number of observations are made before the control charts are first applied, the average level is calculated and the average is plotted on the control chart (Duncan, 1967; Montgomery, 1980). The following procedure is then applied to create the control charts.

1. Check to see that the data (x) meets the specified criteria:
   - Data should usually be distributed around an average.
   - Measurements need to be independent of one another.

2. Find the mean of the group ($\overline{x}$).

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.2}$$

   On the average to delimit the central line:

$$Center\ line = \overline{x} \tag{3.3}$$

3. Calculate the standard deviation ($\sigma_{\overline{x}}$) of the data points.

$$\sigma_{\overline{x}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.4}$$

4. Calculate the upper and lower control limits (UCL, LCL) using the formulas:

$$UCL = \overline{x} + L\sigma_{\overline{x}} \tag{3.5}$$

$$LCL = \overline{x} - L\sigma_{\overline{x}} \tag{3.6}$$

   Generally, for the limit-lines, the L parameter is arbitrarily taken as 3. This is typically called three-sigma control limits.

   Additionaly, in some other monitoring projects (van Griensven et al., 2000; Thomann et al., 2002) warning-limits are applied in the control charts. The upper warning limit (UWL) and the lower warning limit (LWL) become:

$$UWL = \overline{x} + L\sigma_{\overline{x}} \tag{3.7}$$

$$LWL = \overline{x} - L\sigma_{\overline{x}} \tag{3.8}$$

   On L is arbitrarily taken as 2.

5. Graph the X-bar Control Chart, by drawing the *Central line*, the *UCL* and the *LCL* lines and put the data on the control chart as the sample number or time (x-axis) versus measurements (y-axis) as the figure 3.10 showed above.

6. Evaluate the graph to see whether the process is out-of-control, as follows.

   To determine out of control points, and indication of a change in the process is observed (Berthouex, 1989). There are some authors (i.e. Montgomery (2008); Nelson (1984, 1985)) that stablish some criteria to decide when a system is out-of-control.

In this specific case, control charts based on standard solutions are used to detect when it is necessary to clean and calibrate the sensors (NMKL, 1990). The procedure to develop this specific type of control charts is based on the method explained above.

pH, conductivity, LDO and Solitax sensors are controlled with standard solutions. At every visit, these measured values are filled in the chart. In case of spectro::lyser and ammo::lyser, to detect out-of-control situations is based on the analysis of the differences between sensor values and corresponding grab samples measured with a reference method as ISO (2003) describes. Standard methods for this type of control can be found in APHA (1995) and ASTM (1990).

The following steps are the procedure used to develop the control charts based on standard solutions applied.

1. Check to see that the data (x) meets the normal distribution criteria.

2. Calculate difference between the measured value (x_i) and the value of the standard solution (μ).
$$d_i = \mu - x_i \tag{3.9}$$

3. The center line is defined in 0.
$$Center\ line = 0 \tag{3.10}$$

4. Calculate the standard deviation ($\sigma_{\overline{d}}$)
$$\sigma_{\overline{d}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} d_i^2} \tag{3.11}$$

5. Calculate the upper and lower control limits (UCL, LCL) using the formulas:
$$UCL = +L\sigma_{\overline{d}} \tag{3.12}$$
$$LCL = -L\sigma_{\overline{d}} \tag{3.13}$$

   In this application L has been defined as 2.

6. Graph the control chart, by drawing the *Center line*, the *UCL* and the *LCL* lines and put the data on the control chart as the sample number or time (x-axis) versus the differences between measured values and the reference value (y-axis).

7. Evaluate the graph to see whether the process is out-of-control.

It is established that when the system it is out-of-control, a calibration is required. If after several calibrations, the system persists to be out-of-control, it is necessary to send the sensor to the company.

The procedure to develop the control charts based on the differences between sensor values and corresponding grab samples measurements is analogous as the procedure of control charts based on standard solutions.

3.3.5 Chronology of the activities

According to cleaning, calibration and control charts protocols (Sections 3.3.2, 3.3.3 and 3.3.4, respectively), the participants of the project create a schedule including all the activities required each time that a mon*EAU* team goes to the stations.

The manipulations for the cleaning, calibration and control charts are done by at least two people to ensure security and facilitate the process. To control all these activities a maintenance excel file must be filled by the operators (Appendix A). The following activities should be performed at each field visit to keep the sensors operable:

1. Note the date and time (hour:minute) in the excel file *Maintenance* for every sensor and its values.

2. Deactivate the automatic cleaning and despressurize the air supply to the ammo::lyser and spectro::lyser.

3. Remove of any large materials or sediments that have attached to the sensors probe housing.

4. Control the water temperature with a mercury thermometer and fill the excel file *Maintenance* on *Temperature* sheet with this value.

5. Open the metallic cage that is in the water, take out the sensors and clean every sensor.

6. Perform the control chart according to the control charts procedure (Section 3.3.4) for the pH sensor with the standard solutions of pH 7 and 10.

7. Fill the values on *Maintenance* excel file and check if calibration is required.

8. If calibration is necessary, calibrate the sensor as mentioned in the specific SOP (modelEAU, 2012d).

9. Repeat the steps 6, 7 and 8 for the conductivity sensor with a 1000 μS/cm NaCl solution in accordance to the sensor SOP (modelEAU, 2012a).

10. Repeat the steps 6, 7 and 8 for the solitax sensor to verify the accuracy with nano pure water, 200 NTU and 800 NTU solutions in compliance with the SOP of this sensor (modelEAU, 2012e).

11. Repeat the steps 6, 7 and 8 for the LDO sensor with the special bag: air saturated with water (100%) as is mentioned in the particular SOP (modelEAU, 2012c).

12. Pick a sample for the ammo::lyser and spectro::lyser operation verification.

13. Put back the sensors in place in the metallic cage.

14. Activate the automatic cleaning and the air supply.

15. Wait a few minutes for the measurements to stabilize.

16. Write the values after the maintenance procedure in the *Maintenance* excel file.

17. Download the data with the softwares Server, ana::pro and Insight.

18. Analyse the ammonia, nitrate, potassium, COD and TSS in the lab.

19. Fill in the *Maintenance* file with the lab results and check if a calibration is needed for the spectro::lyser and ammo::lyser. If the calibration is demanded, both sensors have to be calibrated during the next visit to the stations following the steps specified in the SOPs for the spectro::lyser (modelEAU, 2012f) and ammo::lyser (modelEAU, 2012b).

3.4 Database

Inside the evaluation process on WQ, the tasks of storage, analysis and interpretation of the collected data during the monitoring period are required. Without a good methodology to store and analyse data, maintaining quality control on the interpretation and the final evaluation is practically impossible (WSDOT, 2008).

These activities are carried out on a system with high graphics capabilities, involving databases, statistical analyses, tendencies determination, correlations, etc.

A database is a structure collection of data. Databases may be stored on a computer and examined using a program. These programs are often called databases, but more strictly they are database management systems (DMS). A database must be built carefully in order to be useful on a computer.

Computer-based databases are usually organised into one or more tables. A table stores data in a format similar to a published table and consist of a series of rows and columns. The main advantage of computer-based tables is that they can be presented on screen in a variety of orders, formats, or according to certain criteria.

In the mon*EAU* project, a WQM database was created to manage and store its data without losing quality. This database is called dat*EAU*base. Furthermore, this system is generalized and it is used to storage data from different research projects conducted as model*EAU*. The main interests to create a database for a huge amount of WQ data are:

- Reducing time spent entering data.
- Providing quality control during data entry.
- Providing centralized storage and retrieval of WQ data.
- Providing reporting functions to assist in data analysis and process improvements.

The developed dat*EAU*base consists in two parts:

- The set linked relational database tables that contain data for a given project(Appendix B.2).
- The user interface, which is a set of Windows-based and R forms, reports, graphs, and auxiliary programs to facilitate data entry, exporting and visualization (Appendix B.5 and B.6).

As the purpose of the database is to store information in a useful way, the database is comprised of multiple tables that contain records and fields. The fields describe the type of information stored, and the records are the items in the database.

Prior to designing the dat*EAU*base a search to identify suitable metadata content standards and data models to leverage was carried out As Sheldon et al. (2011) mentioned, it was observed that most published database designs and metadata standards are oriented concerning documenting measurement details, giving priority to data collection activities and data set characteristics rather than monitoring programs and locations.

However, after evaluating the obtained data from different applications, reviewing some comprehensive databases (i.e. EPA (2012b) database STORET and USGS (2012) database NWIS) and following other procedures to design databases (i.e. Sheldon et al. (2009), EDSC (2006), EPA (2004) and USGS (2002)) a database was created. The advantages of using a database is that it includes data import and export functionalities, data treatment, standard data format, etc.

3.4.1 Database design

The database was developed using Microsoft Office Access 2010. This software is a computer application for DMS. Supported by Microsoft, it combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools.

In order to provide the needed flexibility in the database model, a two-table design is included for organization and monitoring program metadata. The relational diagram for WQ data shown in figure 3.11, accounts descriptions of the primary tables as well as the numerous lookup tables required to define the codes contained in the primary tables.

The primary tables included in dat*EAU*base are WaterQuality, Hydraulics and Weather. The main structure of the primary tables was created as general as possible for any environmental parameter. Table 3.1 shows the contents of the fixed fields for WQ with the appropriate data types and description for storing information entries. In case of Hydraulics and Weather, the relationships between each primary table and the lookup table are analogous.

Table 3.1: Data contained on primary tables.

Field	**Description**	**Format**
Project	Project identifier name	Text
Responsible	Last name identifying the person who managed the data	Text
Site	Location of the sampling area	Text
Sample Point	Specific location on the site where readings were taken	Text
Description	Experiment type description	Text
Experiment Number	Number of replicas for an experiment	Number
Sampling Date	Date on which the WQ, hydraulics and weather readings were taken	Date/Time (dd/mm/yyyy)
Sampling Time	Time at which the WQ, hydraulics and weather readings were taken	Date/Time (hh:mm:ss)
Value	Parameter value	Number
Parameter (unit)	Code identifying parameter name and the abbreviation of the units of the parameter value	Text
Method	Code identifying field/laboratory test procedure	Text
Comments	Comments related to sampled parameter value	Text

Monitoring and testing results have to be identified for some characteristics: location, time, methodology, and other relevant information also must be specified and docu-

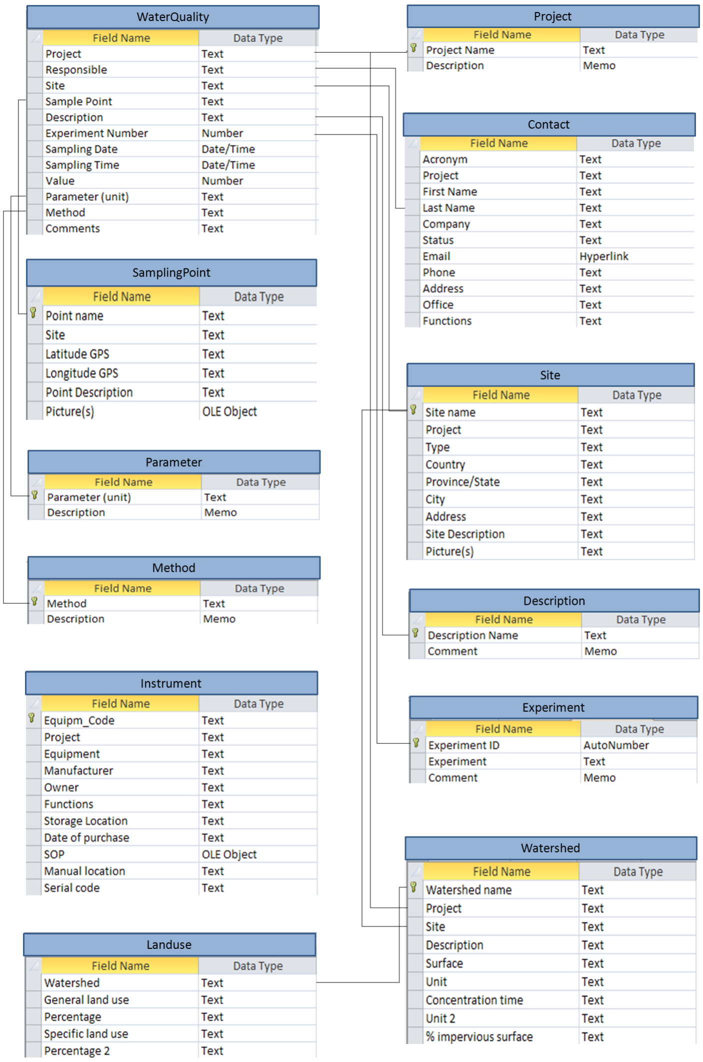


Figure 3.11: Relationships diagram for organization and monitoring program elements of the datEAUbase.

mented. The primary tables contain some fields that are described or defined in detail in other lookup tables. With the relationships between the lookup tables and primary data tables and enforcing referential integrity, data managers are restricted to entering only valid lookup table values into the primary data tables. The lookup tables designed for datEAUbase are: contacts table (Table 3.2), descriptions table (Table 3.3), experiments table (Table 3.4), instruments table (Table 3.5), landuses table (Table 3.6), methods table (Table 3.7), parameters table (Table 3.8), projects table (Table 3.9), sampling points table (Table 3.10), sites table (Table 3.11) and watersheds table (Table 3.12).

Table 3.2: Data contained on contact lookup table.

Field	Description	Format
Acronym	Initials of the First and Last name	Text
Project	Name of the project	Text
First Name	First Name of who is involved in a project	Text
Last Name	Last Name of who is involved in a project	Text
Company	Name of the company who is working on it	Text
Status	Position inside the company	Text
E-mail	E-mail address of the contact	Hyperlink
Phone	Phone number of the contact	Text
Address	Address of the company	Text
Office	Office number	Text
Functions	Which functions the contact has	Text

Table 3.3: Data contained on description lookup table.

Field	Description	Format
Description Name	Experiment type description	Text
Comment	Comments related to the description field	Memo

Table 3.4: Data contained on experiment lookup table.

Field	Description	Format
Experiment ID	Experiment identification	Text
Experiment	Name of the experiment	Text
Comment	Comments related to the experiment	Memo

3.4.2 Data entry and management

According to the database designed in Microsoft Access, an interface has been created with Microsoft Excel helped by Visual Basic for Applicantions (VBA) code.

Microsoft Excel is a spreadsheet application broken up by rows and columns written by Microsoft for Microsoft Windows and Mac OS X, typically used to display and manipulate numerical data. It provides data organization, calculations, graphing tools, pivot tables, and a macro programming language called VBA.

Table 3.5: Data contained on instrument lookup table.

Field	**Description**	**Format**
Equipm_Code	The unique code of the instrument	Text
Project	Project identifier name	Text
Equipment	Name of the instrument used in a project	Text
Manufacturer	Name of the equipment manufacturer	Text
Owner	Owner of the equipment	Text
Functions	Functions of the instrument	Text
Storage Location	Place where the instrument is stored	Text
Date of Purchase	Date and place when and where the instrument was bought	Text
SOP	Standard operation procedure to use the instrument	OLE Object
Manual Location	Place where the manual is stored	Text
Serial Code	Serial code identification of the instrument	Text

Table 3.6: Data contained on land use lookup table.

Field	**Description**	**Format**
Watershed	Name of the watershed	Text
General Land Use	General land use of the watershed	Text
Percentage	Percentage of the general land use of the watershed	Text
Specific Land Use	Specific land use of the watershed	Text
Percentage 2	Percentage of the specific land use of the watershed	Text

Table 3.7: Data contained on method lookup table.

Field	**Description**	**Format**
Method	Method code identifying field/laboratory test procedure	Text
Description	General description of the method	Memo

Table 3.8: Data contained on parameter lookup table.

Field	**Description**	**Format**
Parameter (units)	Parameter code identifying with its abbreviation	Text
Description	General description of the parameter	Memo

Besides Microsoft Excel, VBA is a dialect of Visual Basic implemented by Microsoft Office on its applications. It enables building user defined functions, automating pro-

Table 3.9: Data contained on project lookup table.

Field	**Description**	**Format**
Project Name	Project code identifying	Text
Description	General description of the project	Memo

Table 3.10: Data contained on sampling point lookup table.

Field	**Description**	**Format**
Point Name	Sampling point located on a site	Text
Site	Site name	Text
Latitude GPS	Coordinates GPS - Latitude	Text
Longitude GPS	Coordinates GPS - Longitude	Text
Point description	General description of the sampling point	Text
Picture(s)	Pictures file(s)	OLE Object

Table 3.11: Data contained on site lookup table.

Field	**Description**	**Format**
Site Name	Name of the site	Text
Project	Name of the project	Text
Type	Type of sampling site	Text
Country	Location Country	Text
Province/State	Location Province/State	Text
City	Location City	Text
Address	Address location	Text
Site Description	General description of the site	Text
Picture(s)	Pictures of the site	OLE Object

Table 3.12: Data contained on watershed lookup table.

Field	**Description**	**Format**
Watershed Name	Name of the watershed	Text
Project	Name of the project	Text
Site	Name of the site	Text
Description	Short description of the watershed	Text
Surface	Surface of the watershed	Text
Unit	Abbreviation of the units of surface value	Text
Concentration Time	Time of the concentration of the watershed	Text
Unit 2	Abbreviation of the units of concentration time value	Text
% Impervious Surface	Percentage of the impervious surface	Text

cesses and accessing the Windows API and other low-level functionality through dynamic-link libraries. This tool has been developed to facilitate the data importation from the user without the necessity to open the database for it (Appendix C.1).

Sampling date, sampling time and value are considered measured values. The other fields like project, responsible, site, sample point, description parameter and method are specific descriptions of the measured value called metadata.

3.4.3 Database querying and data export

Finally, a tool to export the data has been created with the software platform R. It allows to open the database, query the data, create graphics and evaluate them (Appendix C.2).

R is an open source software environment and language for statistical computing and graphics. It can be downloaded for free at the web page of this program (R-Project, 2012)).

R provides a wide variety of statistical (linear and non-linear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. It is extensively used among statisticians and data miners for developing statistical software and data analysis. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

R runs on Microsoft Windows platforms using the Object Database Connectivity (ODBC) package. Providing a file name of a data base object or a Data Source Name (DSN) and other connection information, the R software can interact with the database. Access database files can directly interact with R.

Furthermore, R can be extended via packages. These packages are available through the R distribution and the CRAN family of Internet sites covering a very wide range of modern statistics. In this case, to run the created functions, two specific packages are required:

- **ROBC**: This package provides access to databases (including Microsoft Access and Microsoft SQL Server) through an ODBC interface.
- **gWidgetsRGtk2**: This packages provides the possibility to create a GUI. It is a port of gWidgets API to RGtk2. The GTK toolkit is interfaced via the RGtk2 package, which in turn is derived from the RGtk package.

  The packages RGtk2, cairoDevice and gWidgets also are required. These can be installed by following the dependencies for gWidgetsRGtk2.

They have to be installed before running the created functions.

Chapter 4

Results

In this section the database, the import interface, the export interface, some examples of good data, and some examples of control charts are introduced.

4.1 Application of database

As described before (Specifically, sections 1.2.3 and 3.4), user-friendly tools have been developed to manage, treat and evaluate a huge amount of data.

The tools created to achieve the main goal of this thesis, an overview on how to manage them and some examples of their applications are presented in the next sections.

4.1.1 Import interface

An interface has been created by Microsoft Excel and VBA offering an easy-to-use data import to the database as detailed in section 3.4.2. The interface appearance is given in figure 4.1.

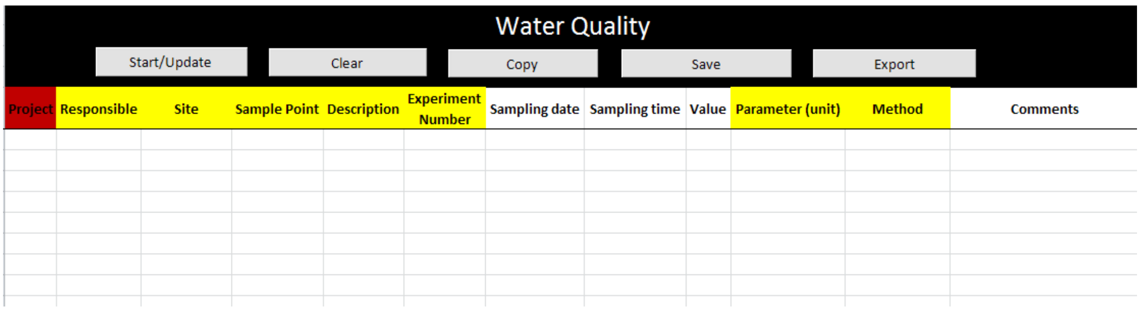


Figure 4.1: Microsoft Excel interface to import the data to the database.

On that interface, several coloured cells and buttons can be perceived. Every colour has a different meaning:

- *Red*: Designates a description field that has to be filled first. Different options are shown on a pick list, one project must be chosen before one can fill the other description fields.
- *Yellow*: Specifies description fields. Each one has to be selected from the specific pick list. The elements of every pick list depend on the project.
- *White*: Indicates the measured values columns. These columns have to be filled by hand.

Moreover, each button at the top of the interface has a different function:

- **Start/Update** button permits to start the selection from the pick lists. On the other hand, if there is any new element added in one of the pick lists, pressing this button will update the list.
- **Clear** button deletes any value on the table.
- **Copy** button copies the first line of description data until the last measured value.
- **Save** button saves the data table in a new excel file without the buttons and the VBA code.
- **Export** button exports the data to the database, after checking the date, time and value format and saves the file.

All codes developed for each function are detailed on the appendix C.1.

After the presentation of the different factors that compound the import interface, the main steps to follow for importing data are presented:

1. Open the table interface.
2. Press the button *Start/Update* when the user is ready to start the data introduction.
3. Choose a project on the first row (Figure 4.2).
4. Only for the first row, complete the other description fields choosing an element on each pick list. As an example, choose a site on its pick list like it is shown in figure 4.3.
5. Insert the data (Figure 4.4).
6. Copy the first row until the last row by pressing *Copy* button (Figure 4.5).
7. Save the data on a table by the *Save* button.
8. Import the data to the database with the *Import* button.

For more information of how to introduce data to the database, how to add elements on the pick list or a project and more relevant instructions, the reader is referred to the user's guide (Appendix B.5).

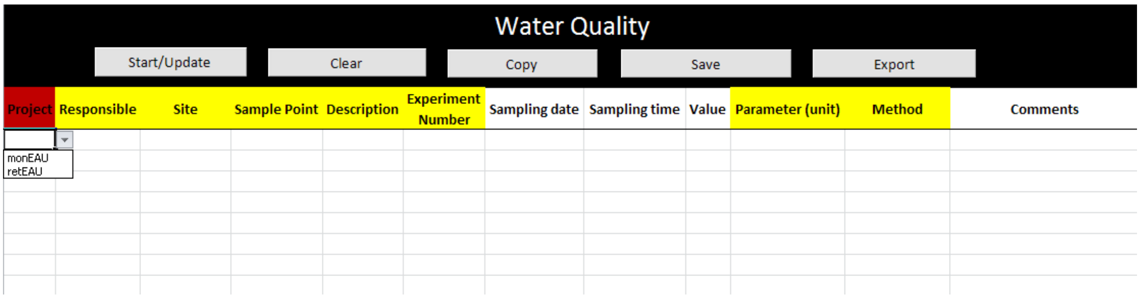


Figure 4.2: Project pick list.

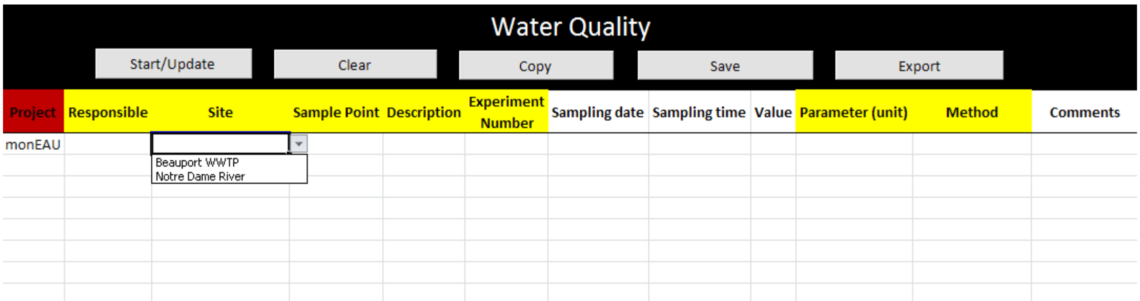


Figure 4.3: Example of site's pick list.

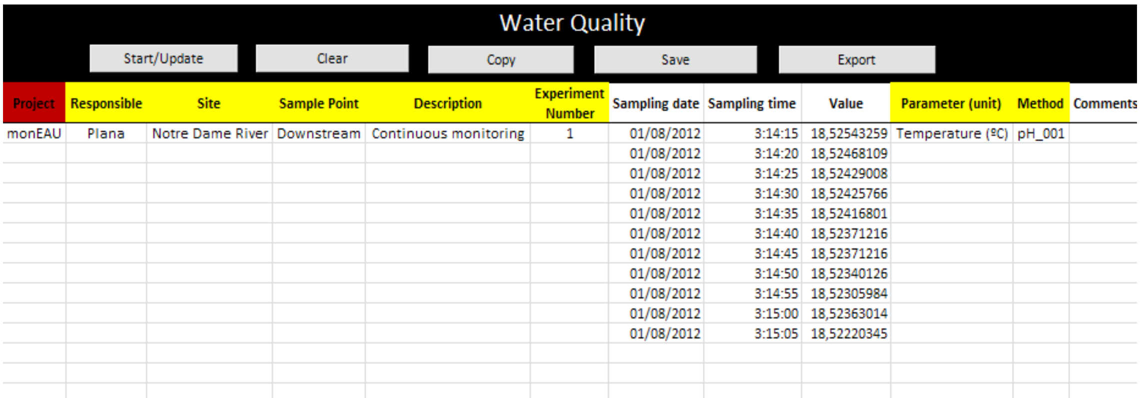


Figure 4.4: Data introduced as example from downstream station.

4.1.2 Export interface

As mentioned in section 3.4.3 characterises, several tools have been developed to request data from the database and make graphics to evaluate the data. To this end, three main functions have been created to make these tools more user-friendly:

Water Quality

Start/Update | Clear | Copy | Save | Export

Project	Responsible	Site	Sample Point	Description	Experiment Number	Sampling date	Sampling time	Value	Parameter (unit)	Method	Comments
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:15	18,52543259	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:20	18,52468109	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:25	18,52429008	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:30	18,52425766	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:35	18,52416801	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:40	18,52371216	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:45	18,52371216	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:50	18,52340126	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:14:55	18,52305984	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:15:00	18,52363014	Temperature (ºC)	pH_001	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	3:15:05	18,52220345	Temperature (ºC)	pH_001	

Figure 4.5: Table ready to be saved and imported to the database.

- **Open the database function**: It permits to open a table from a database. It displays the opened data table.
- **Query data function**: It is a query data interface. The output is the data selected to be plotted.
- **Plot data function**: It allows to make scatterplots of up to four series of data.

To execute these functions, next steps must be followed:

1. Define the directory.

   The command to set a directory in R is:

```
setwd(dir)      # dir is a character string and it defines the directory
    path.
```

2. Read the function to open the database developed in Microsoft Access.

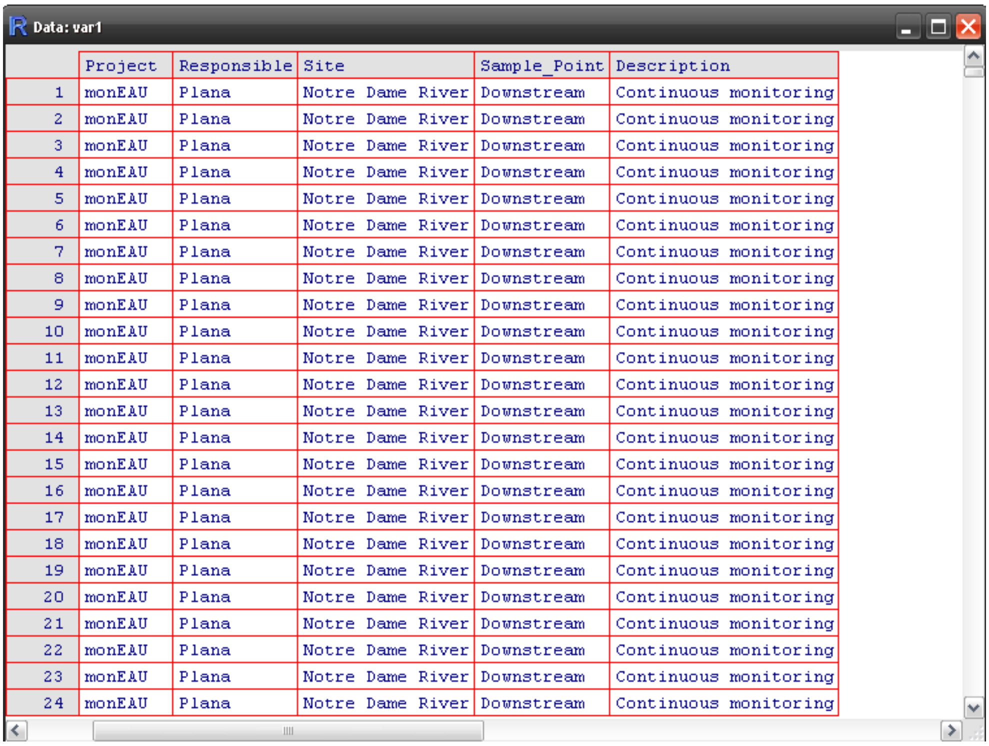
```
source("opendb.R")
```

3. Execute the opening database function.

```
opendb("file", "tab")
```

   file and *tab* are the inputs of this function. On the *file* input, the filename of the database has to be indicated and on the *tab* input, requires the name of the table from the database to be opened.

4. A table with the exported data is displayed (i.e. figure 4.6).



	Project	Responsible	Site	Sample_Point	Description
1	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
2	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
3	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
4	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
5	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
6	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
7	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
8	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
9	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
10	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
11	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
12	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
13	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
14	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
15	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
16	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
17	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
18	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
19	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
20	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
21	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
22	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
23	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
24	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring

Figure 4.6: Table displayed after exporting the data from the database.

5. Read the function *doit*. It is a subfunction used on the *query* function.

```
source("doit.R")
```

6. Read the function to query the data to be plotted.

```
source("query.R")
```

7. To execute this function the following instruction has to be ordered:

```
query(Dat1, Dat2)
```

   Dat1 and *Dat2* are the inputs of this function. These inputs mean *Start Date* and *End Date* to delimit the period of data to be plotted. The format of the inputs must be "YYYY-MM-DD".

8. After this a window is displayed (Figure 4.7), showing several pick lists depending on the data exported from the dat*EAU*base.

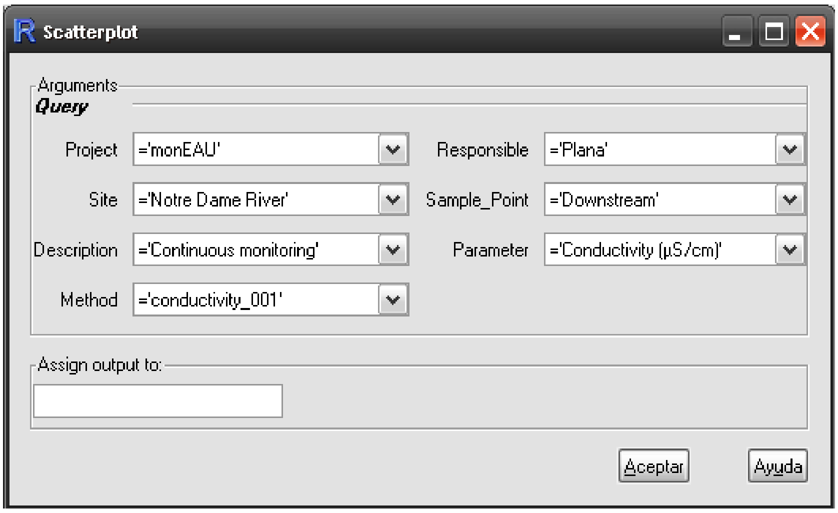


Figure 4.7: Window displayed to query the data.

9. Choose an element from each pick list to select which parameter is going to be plotted (i.e. figure 4.8).

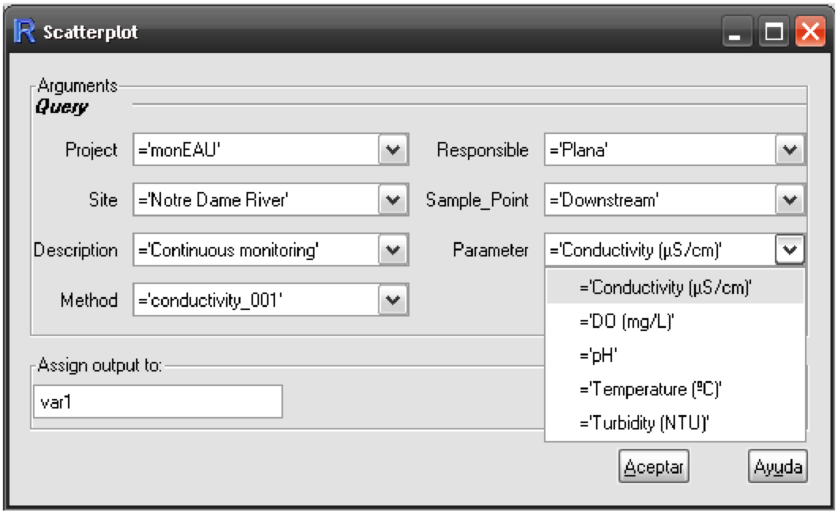


Figure 4.8: Example of pick list to query the data.

10. Fulfil the *Assign output to* field to designate a name to the selected variable (Figure 4.9).

    It is suggested to name the variables var1, var2, ... in numerical order depending

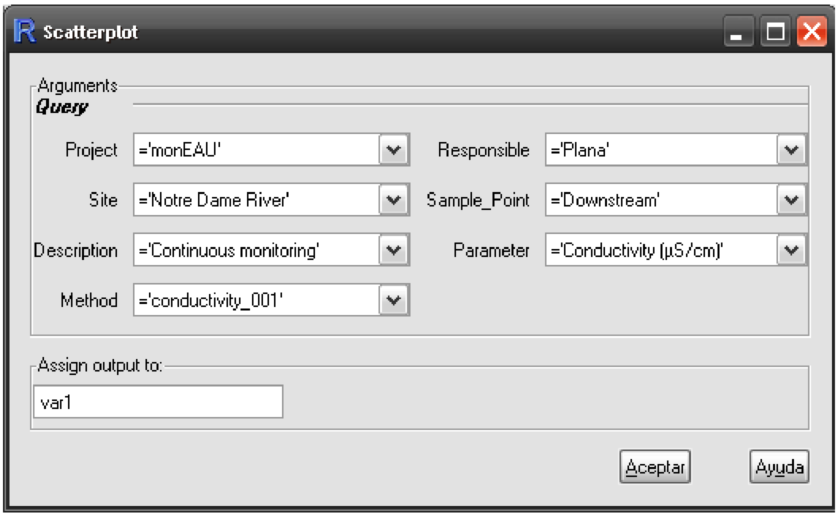


Figure 4.9: Example of pick list to query the data.

on how many variables are going to be plotted. Moreover, after assigning the variable name, it is sufficient to only press the *accept* button before to defining and denominating the next variable it is sufficient.

11. Press *accept* button.

12. Read *plotgraph* function.

```
source("plotgraph.R")
```

13. Execute *plotgraph* function.

```
plotgraph(n, m)
```

Where n means the number of variables defined on the query function to be plotted in the same graphic. And m means how many axes want to be plot.

Only there is the possibility two axis graph when two parameters are going to be plotted.

14. A scatterplot is displayed.

The graph is automatically saved as *graph.png* to the set directory. Before making another graphic, its name must be changed.

For more information on how to request the data, plot them and more relevant instructions, the reader is referred to the details in the user's guide (Appendix B.6).

4.1.3 Examples of good data

The dat*EAU*base is particularly useful to manage a huge amount of data permitting to keep the data in the same format in one document and, the most important point, to keep its quality. Moreover, the user-friendly tools accompanied by GUIs allow to manage this high quality data easily.

The result of this design and the developed tools described above, can be illustrated with some graphics. Detailed in section 4.1.2, the *plotgraph* function can make graphics for up to four different variables. Below, some examples are given:

- Plot one parameter from one sensor. For example, the temperature measured by conductivity sensor at the downstream station (Figure 4.10).

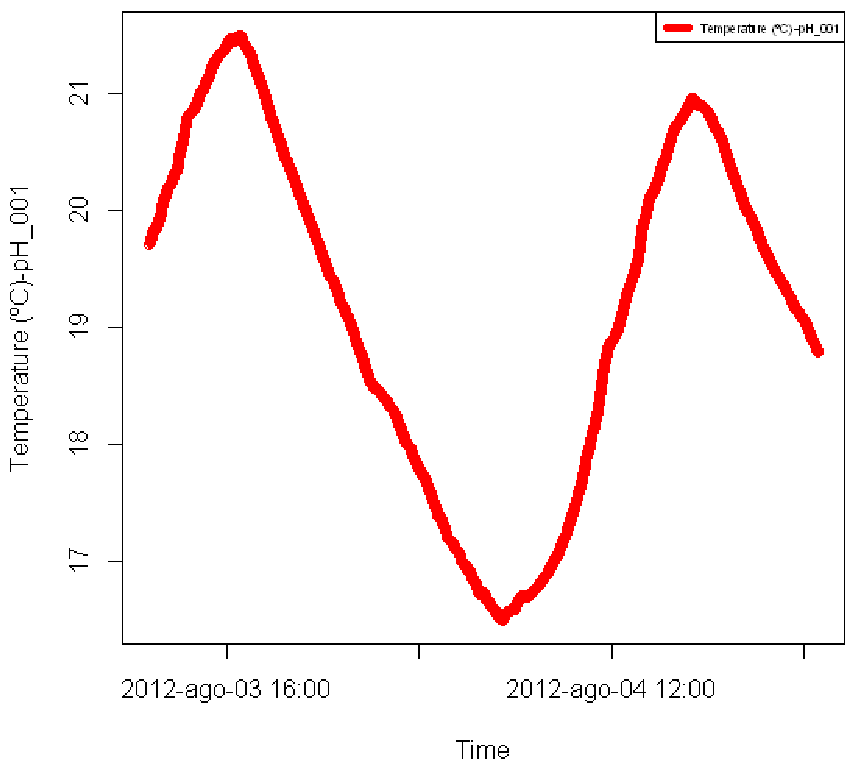


Figure 4.10: Temperature graph measured by conductivity sensor at downstream station.

- Plot one parameter measured by different sensors. For example, the temperature measured by the conductivity, LDO and pH sensors at the downstream station (Figure 4.11).

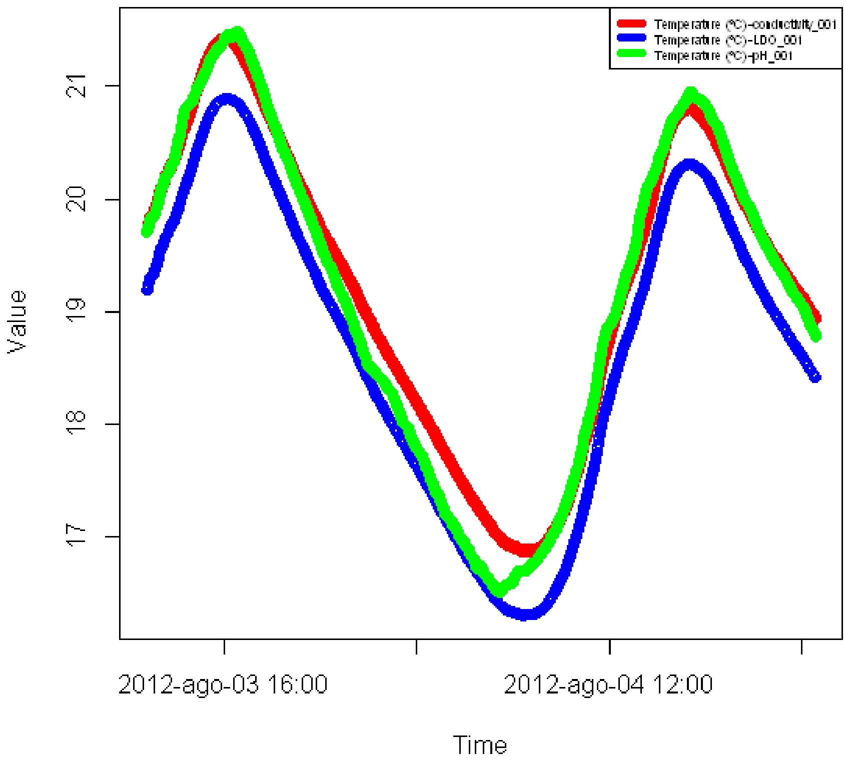


Figure 4.11: Temperature graph measured by conductivity, LDO and pH sensors at downstream station.

- Plot one parameter measured by one sensor at both stations. For example, the temperature measured by the conductivity sensor at both stations (Figure 4.12).
- Plot several parameters measured with the same sensor. For example, both pH and temperature measured by the pH sensor at the downstream station (Figure 4.13).

In conclusion, besides keeping high quality data, these tools allow to compare values between sensors, stations, different periods of data or any comparison that is desired as it is illustrated in the graphics above.

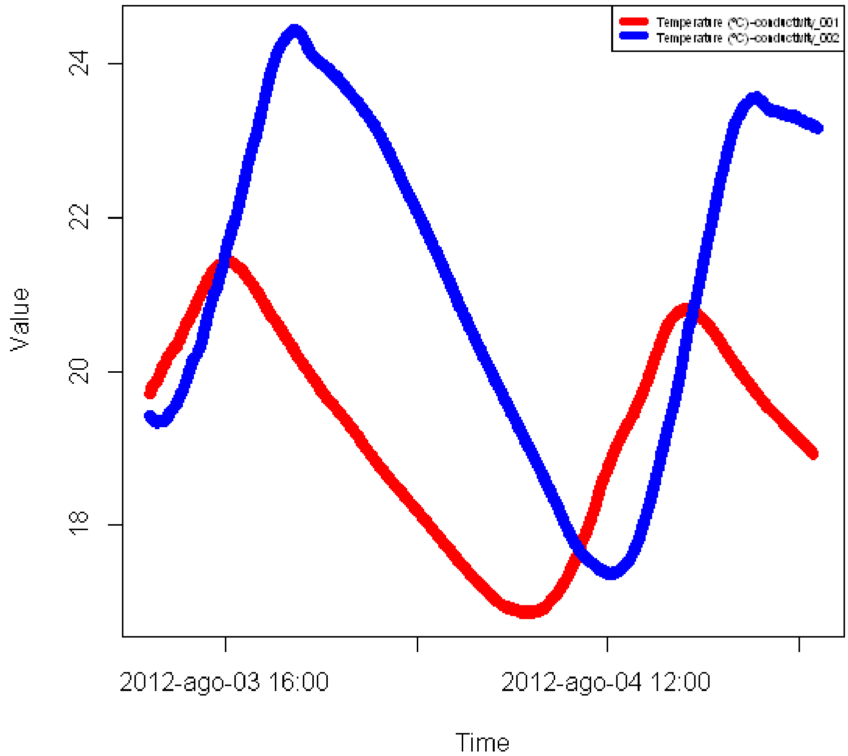


Figure 4.12: Temperature graph measured by conductivity sensor at both stations.

4.2 Application of control charts

Following the steps specified at the section 3.3.4 the control charts were created considering the frequency of checking the control charts twice per week for this installation.

Below two different options to build control charts and some examples of control charts are showed.

4.2.1 Modes to build the control charts

Two different kinds of control charts depending on the limits are set out:

1. *Control charts with fixed limits*: Using a few good values to fix the limits.

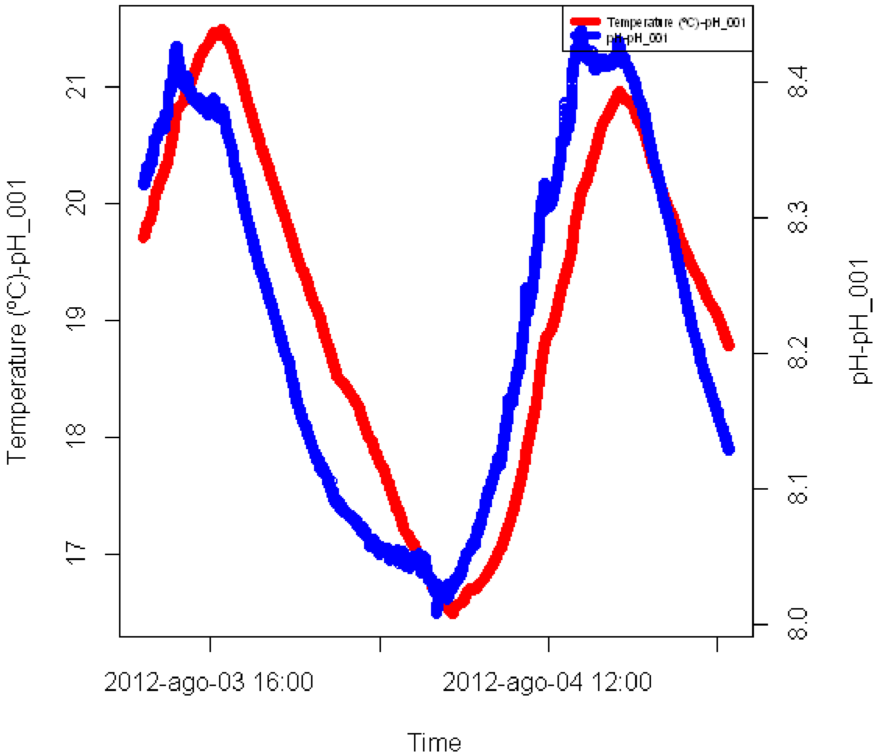


Figure 4.13: Temperature and conductivity graph measured by the pH sensor at downstream station.

Dismissing outliers, fixed limits permit calculating enclosed restrictions assuring a good reliability to detect when the process is out-of-control.

2. *Control charts with variable limits*: Each value obtained is used to continuously recalculate the limits.

In the better approach, each value measured every time that the station is visited, it is included to recalculate the limits. If the value obtained is poor, it has not to be considered.

In that case, there are two possible scenarios: In the first one, the acceptability zone of in-control situations is decreasing in the course of the study, but on the other hand, in the second one, the acceptability zone of in-control situations is increasing along the measurement period.

Considering that the interest of the control charts is to detect out-of-control situations to clean and calibrate the sensors, it is necessary for the limits not to be excessively permissible. Furthermore, the variable limits can be effected drastically by an outlier in case of an insufficient number of measured values.

Summing up, to assure reliability in detecting when it is necessary to clean or calibrate the sensor and control how the sensor is working, control charts with fixed limits are used in this implementation.

In the next figures the two types of control charts used in this project are given. The first one is a control chart based on standard solutions (Figure 4.14) and the other one is a control chart based on the difference between sensor values and concentration values found in grab samples measured into the laboratory (Figure 4.15). In these applications, to fix the control limits were fixed on the basis of one month of acceptable values.

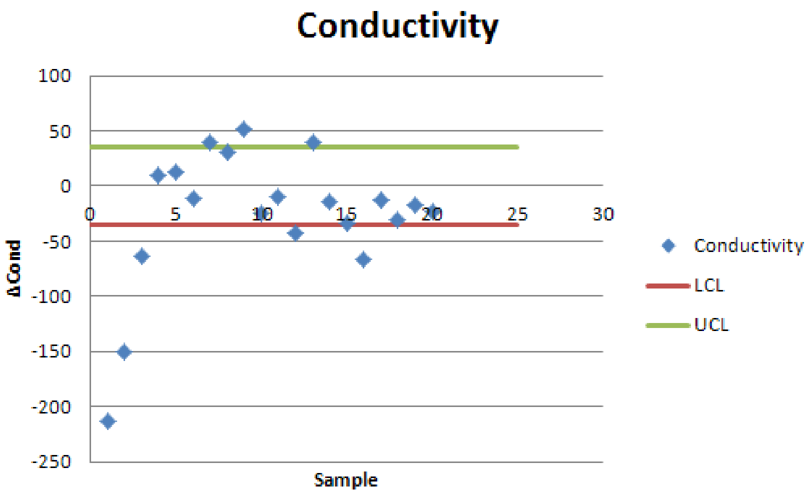


Figure 4.14: Control chart for conductivity with fixed limits (mon$EAU1$).

4.2.2 Examples of control charts

In this section, some examples of control charts built for some parameters in this project are presented. Observing when the sensors have been calibrated or replaced, the behaviour of the sensors can be illustrated.

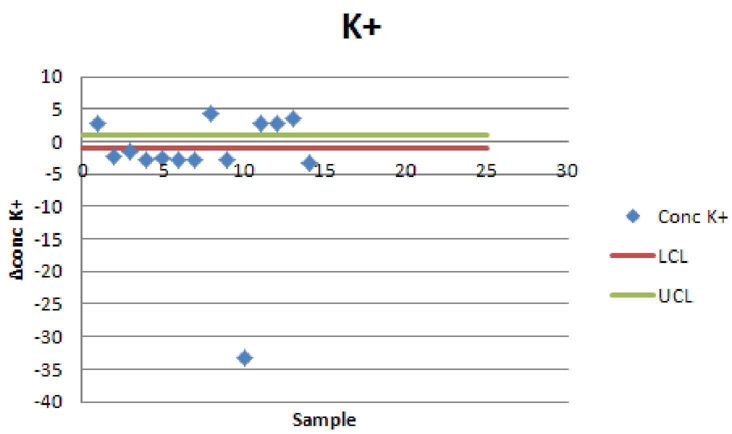


Figure 4.15: Control chart for potassium with fixed limits (mon$EAU1$).

Conductivity sensor

Figure 4.16 shows the conductivity control chart from mon$EAU1$. It is a good example of a control chart based on standard solutions.

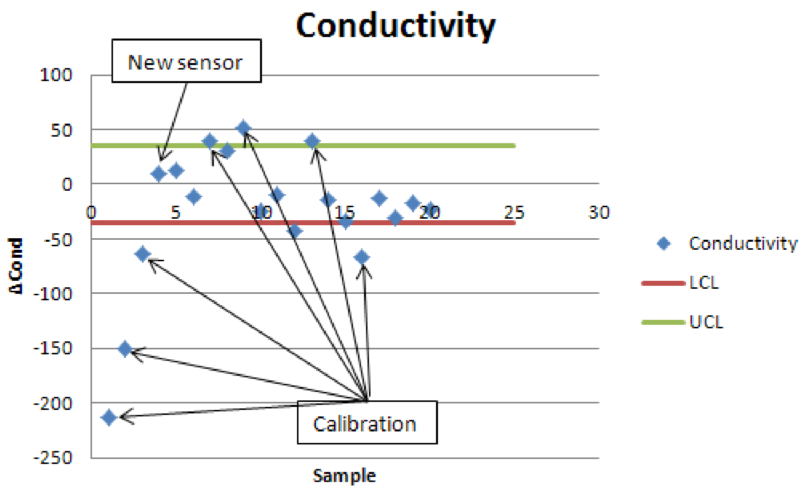


Figure 4.16: Control chart for conductivity with variable limits (mon$EAU1$).

After the start up of the station, the sensor was not working well. Following several calibrations, it was getting better but all situations were out-of-control. Succeeding that, the sensor was replaced. Afterwards, it has been working correctly and it was calibrated when ΔCond was outside the control limits.

pH sensor

To control how the pH sensor is working, two control charts for the pH were developed, one with the standard solution at pH 7 (Figure 4.17), and the other with the standard solution at pH 10 (Figure 4.18). Additionally, since the pH sensor also measures the temperature, another control chart was elaborated for this parameter with the difference between the sensor value and the corresponding thermometer value (Figure 4.19).

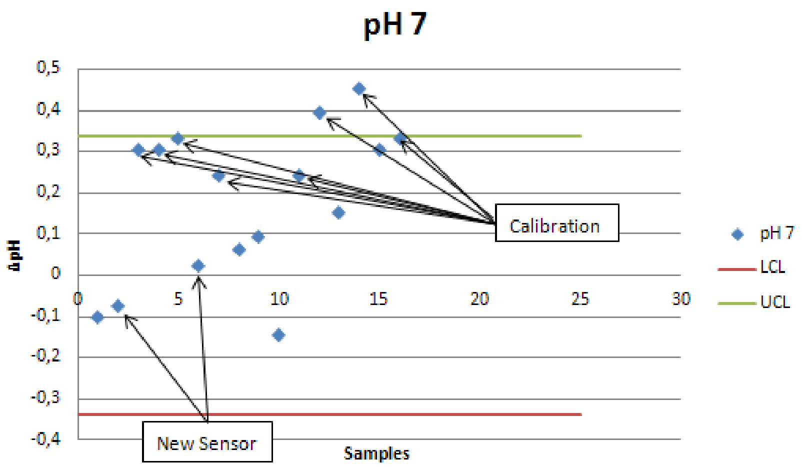


Figure 4.17: Control chart of pH 7 for pH sensor from monEAU1 with fixed limits.

At the beginning of the installation operation, all calibrations failed. Moreover, even the temperature values were outside of control limits. To solve it, the cap of this sensor was replaced twice and also filled with new electrolyte liquid.

After the second replacement of the cap, it has been working properly and it has been calibrated every time one value of one of these three control charts was out-of-control. Also, it can be observed that the pH sensor is sensitive: it requires weekly calibration and after each rain event.

Generally, a high error is observed in the pH sensor due to the age of this sensor. After a year working, the sensor cannot work at 100%.

Ammo::lyser

To check when a calibration is needed or how the ammo::lyser is working, control charts for temperature, pH and potassium parameters are used.

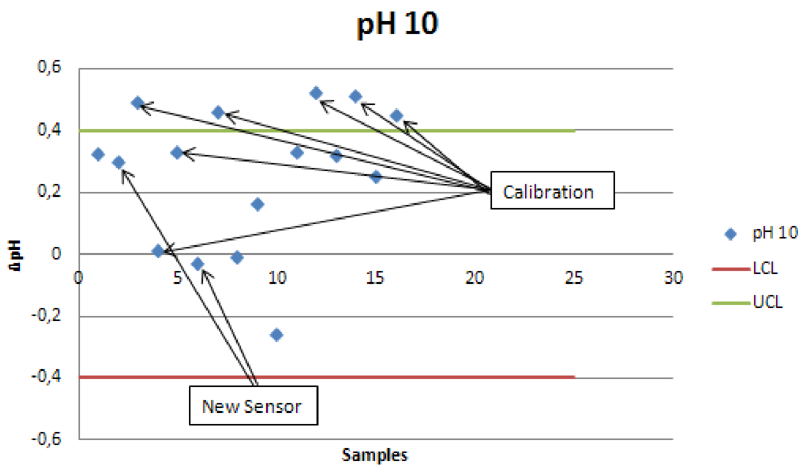


Figure 4.18: Control chart of pH 10 for pH sensor from mon$EAU1$ with fixed limits.

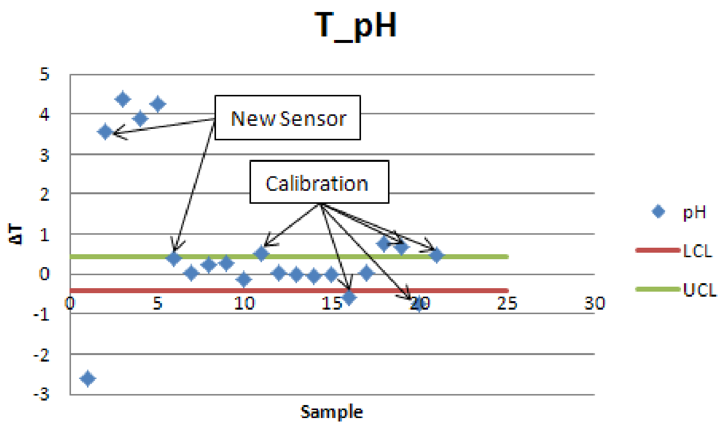


Figure 4.19: Control chart of temperature for pH sensor from mon$EAU1$ with fixed limits.

In the case of temperature (Figure 4.20), its control chart is based on the difference between the sensor values and the corresponding values measured with a thermometer. For the first half of the values, it is observed that the sensor is working inaccurately: generally the ammo::lyser measurements present a bias below the values measured with the thermometer.

Meanwhile, the control chart for pH is based on the difference between ammo::lyser values and corresponding pH sensor values. It is shown in figure 4.21. Comparing two

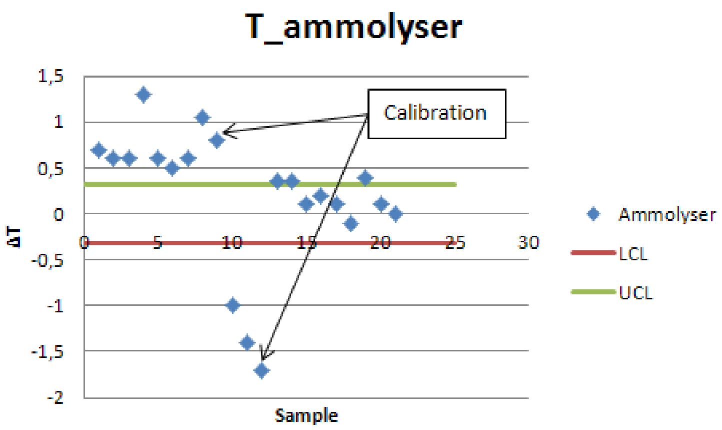


Figure 4.20: Control chart of temperature for ammo::lyser sensor from mon$EAU2$ with fixed limits.

values from different sensors, this control chart is not as important as the temperature and potassium control charts. Moreover, it has to be interpreted in a different manner than the other control charts, depending on how the pH sensor is working. After checking how the pH sensor is working, the following situations can be possibles:

1. If the pH sensor is working correctly:
   a) When the difference value is inside the range, the pH electrode of the ammo::lyser is working correctly.
   b) When the difference value is outside the range, the pH electrode of the ammo::lyser is not working correctly.
2. If the pH sensor is not working correctly:
   a) When the difference value is inside the range, the pH electrode of the ammo::lyser is not working correctly.
   b) When the difference value is outside the range, it is not possible to know if the pH electrode of the ammo::lyser is working correctly or not.

In case the pH sensor is working correctly it is calibrated when required. Moreover, since the pH sensor and ammo::lyser are working similarly, it can be supposed that the pH electrode of the ammo::lyser is working correctly as well.
In that case, for a control chart with a reference value, a portable sensor can be used and/or a measurement of the pH in the lab is recommended.

Finally, for potassium, the control chart is based on the analysis of the difference between sensor values and corresponding grab samples values measured with a reference

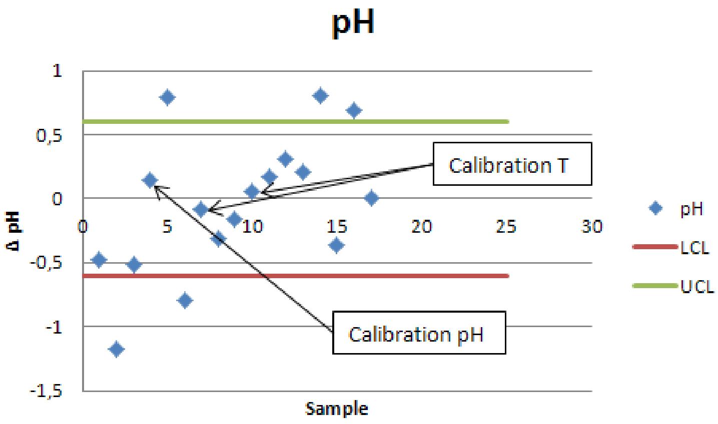


Figure 4.21: Control chart of pH for ammo::lyser sensor from mon$EAU2$ with fixed limits.

method. This cart is presented in the figure 4.22. In this last control chart for potassium, the limits are acceptable when comparing with the magnitude of the values. That is caused by the inaccurate operation of the sensor when the control charts were developed.

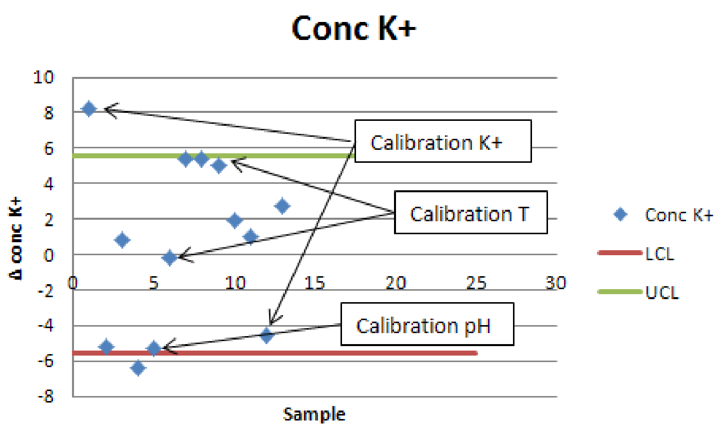


Figure 4.22: Control chart of potassium for ammo::lyser sensor from mon$EAU2$ with fixed limits.

As described in the section 3.2.3 and also, as observed in figures 4.20, 4.21 and 4.22, each parameter has to be calibrated individually and when only a single parameter requires calibration, it is not necessary to recalibrate the other parameters.

The control chart for ammonia has not been developed, because the measurements are of too poor quality and there is not enough data. The selective electrode to measure the ammonia is working inaccurately, and several calibrations with a known NH_4 concentration solution failed. Afterwards, the ammo::lyser has only been calibrated with potassium, temperature and pH.

Solitax

Even though the solitax has been calibrated by the manufacturer, a control chart is needed to verify when a deep cleaning is necessary. Moreover, building several control charts with different concentrations has a special interest because it is possible to observe at which concentration the sensor is more sensitive and more precise.

Three control charts are presented: the control chart band the control chart with 800 NTU as standard solutionased with nano water as standard solution (Figure 4.23), the control chart with 200 NTU as standard solution (Figure 4.24), and the control chart with 800 NTU as standard solution (Figure 4.25).

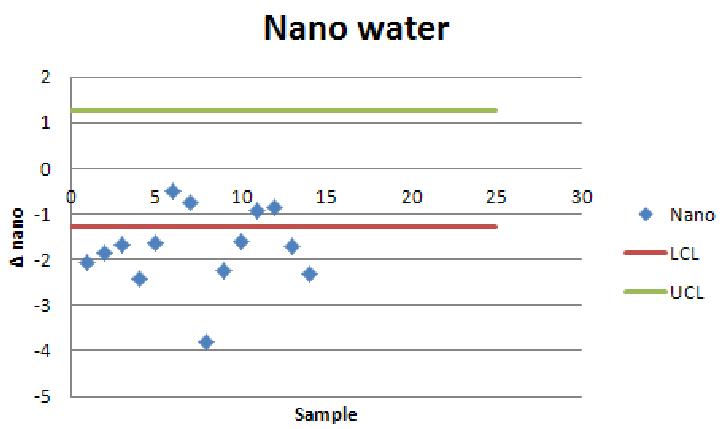


Figure 4.23: Control chart of nano water for solitax from mon$EAU2$ with fixed limits.

As can be noticed on the figures 4.23, 4.24 and 4.25, at lower concentrations of TSS, the solitax is less sensitive and the measurements are less precise.

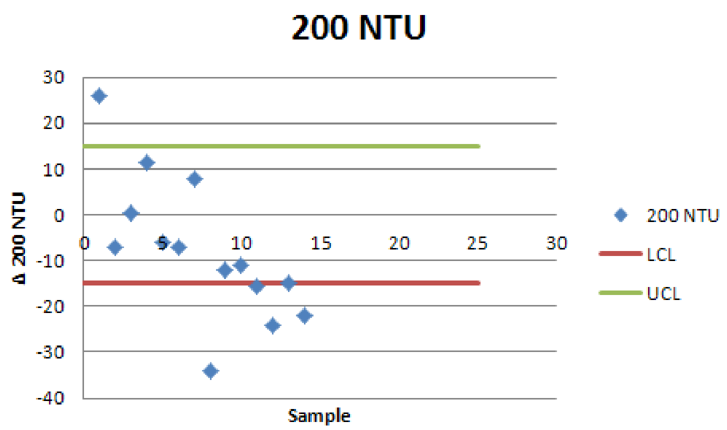


Figure 4.24: Control chart of 200 NTU water for solitax from mon$EAU2$ with fixed limits.

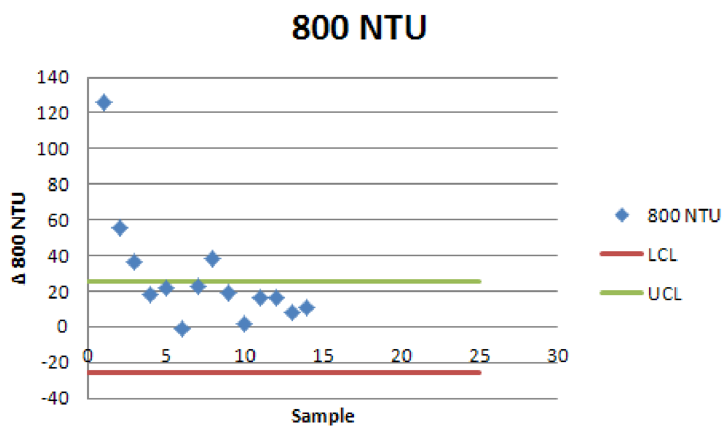


Figure 4.25: Control chart of 800 NTU water for solitax from mon$EAU2$ with fixed limits.

Chapter 5

Conclusions

The main conclusions of this thesis can be split in two main parts: the application of the database and the application of the control charts.

A database for environmental measured values with user interfaces to introduce and query data has been developed. Moreover, to test these user-friendly tools, some data collected in mon*EAU* project were used. The following conclusions have been drawn:

First: A database is greatly useful to manage a huge amount of data permitting to keep the data in the same format in one document and keep its quality.

Second: The structure of dat*EAU*base can be used for any environmental parameter because as it has been designed as general as possible.

Third: Having a database in a research group is very advantageous since there is the facility to look up the desired data from any project and the data remain available and documented.

Forth: The dat*EAU*base created is for rawdata. Even though, in the global mon*EAU* idea, different database can coexist to store treated data, filtrated data, etc. The procedure to create, maintain and manipulate data on these other databases is analogous.

Fifth: The capacity Access is 2 GB per file. In case of mon*EAU* stations the amount of collected data magnitude is extraordinary when taking measures every five seconds, and this capacity is not enough to store all data in one document. Linking some database files will be needed.

Sixth: An import interface created in Excel together with VBA is an easy way to introduce the data to the database. It permits a high speed and an efficiently data import. However, the command *Copy* on the import interface works slowly when applied to a considerable amount of data.

Seventh: The table size to be opened in R has to be less than 1 GB. Therefore, it is another limitation to treat and manage an amount of data as big as the data sets in the mon*EAU* project.

Eighth: The *opendb* function developed in R takes a long time to be executed. The time data from the database is exported in a wrong format and this function has to rearrange it line by line before it can be used.

On the other hand, to assure a high quality data, controlling how the station is working is required. The following conclusions are drawn:

Ninth: Control charts are a very useful system to verify sensors and control how they are working.

Tenth: Control charts permit to check the good operation of the sensors and then assure that the obtained data is reliable and with high quality.

Eleventh: Maintaining the sensors frequently and continuously is essential to keep the station functioning properly even though accessing the stations is complicated.

Chapter 6

Recommendations

The following recommendations for future database projects are proposed:

1. Create a new database with sufficient capacity to store a huge amount of data as in case of mon*EAU* project. A good option could be the SQL database.

2. Develop other user-friendly tools to manage a wide number of data accompanied with a GUI, for example, using C++ language.

   Moreover, below some recommendations are presented to improve the stations' data quality control:

3. Carrying out tests more often than twice per week to allow creating control charts that are more reliable and provide thorough control of the sensors.

4. The control chart of the pH sensor on the ammo::lyser can be performed by measuring the pH of the water in-situ by a portable pH-meter.

5. The control charts of the conductivity sensor and the solitax can be performed by measuring the conductivity and the turbidity by portable sensors.

Bibliography

APHA (1995). *Standard methods for the examination of water and wastewater*. Number v. 9, pt. 1946. American Public Health Association.

ASTM (1990). *ASTM Designation: D 3864 - 79 (reapproved 1990). Standard Guide for Continual On-Line Monitoring Systems for Water Analysis*. Philadelphia.

Bartram, J. and R. Ballance (1996). *Water Quality Monitoring - A Practical Guide to the Design and Implementation of Freshwater Quality Studies and Monitoring Programmes*. United Nations Environment Programme and the World Health Organization.

Beck, M. B., J. B. Watts, and S. Winkler (1998). An environmental process control laboratory: At the interface between instrumentation and model development. *Water Science and Technology 37*(12), 353–362.

Berthouex, P. M. (1989). Constructing control charts for wastewater treatment plant operation. *Research Journal of the Water Pollution Control Federation 61*(9/10), 1534–1551.

Bertrand-Krajewski, J.-L., S. Barraud, G. L. Kouyi, A. Torres, and M. Lepot (2007). Event and annual tss and cod loads in combined sewer overflows estimated by continuous in situ turbidity measurements. *Proceedings of the 11th International Conference on Diffuse Pollution, Belo Horizonte, Brazil*, 26–31.

Bols, J., P. Goethals, K. Meirlaen, A. van Griensven, V. Vandenberghe, L. Van Vooren, N. De Pauw, P. Vanrolleghem, and W. Bauwens (1999). Automated measurement stations for river water quality monitoring. Proceedings 13th Forum Applied Biotechnology. Med. Fac. Landbouww. Univ. Gent, 64/5a, 107-110.

Chapman, D. V. (1998). *Water Quality Assessment: A Guide to the Use of Biota, Sediments and Water in Environmental Monitoring* (2nd ed.). Abingdon ; New York : Taylor & Francis.

Corbitt, R. (1990). *Standard handbook of environmental engineering* (2nd ed.). Blacklick, Ohio, U.S.A.: McGraw-Hill.

Dandy, G. C. and S. F. Moore (1979). Water-quality sampling programs in rivers. *Journal of the Environmental Engineering Division-Asce 105*(4), 695–712.

Duncan, A. (1967). *Quality control and industrial statistics*. Chicago, U.S.A : R. D. Irwin.

EC (2012, June). Monitoring. http://www.ec.gc.ca/. Environment Canada.

EDSC (2006, January). *Environmental Sampling, Analysis and Results Data Standards.* Environmental Data Standards Council.

EPA (2004, January). *Water Quality Database. Database Design and Data Dictionary.* U.S. Environmental Protection Agency.

EPA (2012a, June). An introduction to water quality monitoring. http://water.epa.gov. U.S. Environmental Protection Agency.

EPA (2012b, July). Storet legacy data center. http://water.epa.gov. U.S. Environmental Protection Agency.

European Parliament, C. (2000). Directive 2000/60/ec of the european parliament and of the council of 23 october 2000 establishing a framework for community action in the field of water policy.

Hach (2006a, December). *LDO Dissolved Oxygen Sensor. User Manual* (6 ed.). Hach.

Hach (2006b, October). *pHD sc Digital. Differenctial pH/ORP Sensors* (4 ed.). Hach.

Hach (2008a, September). *3700sc Digital Conductivity Sensor. User Manual* (5 ed.). Hach.

Hach (2008b, November). *SIGMA 950 Flow Meter* (5 ed.). Hach.

Hach (2009, December). *SOLITAX sc. User Manual* (4 ed.). Hach.

Hach (2012a, July). Insight. data management software. http://www.hachflow.com/.

Hach (2012b, July). Sigma 950 flowmeter. http://www.hachflow.com/.

Harmancioglu, N. B. (1997). The need for integrated approaches to environmental data management. *Integrated Approach to Environmental Data Management Systems 31*, 3–14.

Harmancioglu, N. B., M. N. Alpaslan, and V. P. Singh (1998). Needs for environmental data management. *Environmental Data Management 27*, 1–12.

Harmancioglu, N. B. and N. Alpaslan (1992). Water quality monitoring network design: A problem of multi-objective deicions making. *Journal of the American Water Resources Association 28*(1), 179–192.

Harmancioglu, N. B., O. Fistikoglu, S. D. Ozkul, V. P. Singh, and M. N. Alpaslan (1999). *Water Quality Monitoring Network Design.* Water Science and Technology Library. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Helmer, R. (1994). Water quality monitoring - national and international approaches. *Hydrological, Chemical and Biological Processes of Transformation and Transport of Contaminants in Aquatic Environments 219*(219), 3–17.

ISO (1990, March). *ISO 8466-1 Water quality - Calibration and evaluation of analytical methods and estimation of performance characteristics. Part 1: Statistical evaluation of the linear calibration function.* Geneva, Switzerland: International Organization for Standardization.

ISO (2003, October). *ISO 15839 Water quality - On-line sensors/analysing equipment for water - Specifications and performance tests.* Geneva, Switzerland: International Organization for Standardization.

ISO (2006, January). *ISO 5667-1 Water Quality Sampling. Part 1: Guidance on the design of sampling programmes and sampling techniques.* Geneva, Switzerland: International Organization for Standardization.

Karpuzcu, M., S. Senes, and A. Akkoyunlu (1987). Design of monitoring systems for water quality by principal component analysis a case study. In *Proceeding, INT, Symp. On Environmental Management (Environment)*, Volume 87, pp. 673–690.

Lovett, G. M., D. A. Burns, C. T. Driscoll, J. C. Jenkins, M. J. Mitchell, L. Rustad, J. B. Shanley, G. E. Likens, and R. Haeuber (2007). Who needs environmental monitoring? *Frontiers in Ecology and the Environment 5*(5), 253–260.

modelEAU (2011a). *Standard Operating Procedure - Cleaning of the monEAU station sensors.* Université Laval.

modelEAU (2011b). *Standard Operating Procedure - Protocol of configuration and functioning of Sigma 950 Flow Meter.* Université Laval.

modelEAU (2012a). *Standard Operating Procedure - Cleaning and Calibration Procedure fo the Conductivity Sensor.* Université Laval.

modelEAU (2012b). *Standard Operating Procedure - Cleaning and Calibration Procedure of the Ammo::lyser Sensor.* Université Laval.

modelEAU (2012c). *Standard Operating Procedure - Cleaning and Calibration Procedure of the LDO Sensor.* Université Laval.

modelEAU (2012d). *Standard Operating Procedure - Cleaning and Calibration Procedure of the pH Sensor.* Université Laval.

modelEAU (2012e). *Standard Operating Procedure - Cleaning and Calibration Procedure of the Solitax Sensor.* Université Laval.

modelEAU (2012f). *Standard Operating Procedure - Sampling and Calibration Procedure of the Spectro::lyser Senor.* Université Laval.

Montgomery, D. (2008, April). *Introduction to Statistical Quality Control* (6th ed.). Wiley Desktop Editions. New York City, U.S.A.: John Wiley & Sons.

Montgomery, D. C. (1980). The economic design of control charts - a review and literature survey. *Journal of Quality Technology 12*(2), 75–87.

Nelson, L. S. (1984). The shewhart control chart - tests for special causes. *Journal of Quality Technology 16*(4), 237–239.

Nelson, L. S. (1985). Interpreting shewhart xbar control charts. *Journal of Quality Technology 17*(2), 114–11.

NMKL (1990). *Quality Assurance Principles for Chemical Food Laboratories.* NMKL rapport. Oslo, Norway: Nordic Council of Ministers.

Primodal (2012, July). Primodal rsm30. http://www.primodal.com/.

R-Project (2012, September). R-project. http://www.r-project.org/.

Rieger, L. and P. A. Vanrolleghem (2008). moneau: a platform for water quality monitoring networks. *Water Science and Technology 57*(7), 1079–1086.

Sanders, T. (1983). *Design of Networks for Monitoring Water Quality.* Water Resources Publications.

S::can (2006). *Manual ana::pro* (Version 5.3 ed.). S::can.

S::can (2007a). *ammo::lyser V1 Manual* (1 ed.). S::can.

S::can (2007b). *Manual s::can spectrometer probe* (1 ed.). S::can.

S::can (2012). S::can. http://www.s-can.at/.

Sheldon, W. J., C. Laporte, J. Carpenter, and M. Alber (2009). Southeast coastal water quality monitoring metadata tools - database and web applications. Natural resource report, National Park Service, Washington, U.S.A.

Sheldon, W. M., C. Laporte, T. Douce, and M. Alber (2011). *A coastal water quality metadata database for the Southeast U.S.A.*, Chapter 6.12. Warnell School of Forestry and Natural Resources The University of Georgia, Athens, U.S.A.

Thomann, M., L. Rieger, S. Frommhold, H. Siegrist, and W. Gujer (2002). An efficient monitoring concept with control charts for on-line sensors. *Water Science and Technology 46*(4-5), 107–116.

Thomas, O. and M.-F. Pouet (2005). Wastewater quality monitoring: On-line/on-site measurement. In *Water Pollution*, Volume 2 of *The Handbook of Environmental Chemistry*, pp. 211–226. Springer Berlin / Heidelberg.

USGS (2002). *Environmental Database For Water-Quality Data. For the Penobscot River, Maine: Design Documentation and User Guide.* USEPA (U.S. Environmental Protection Agency).

USGS (2012). National water information system (nwis). http://www.usgs.gov/.

van Griensven, A., V. Vandenberghe, J. Bols, N. De Pauw, P. Goethals, J. Meirlaen, P. Vanrolleghem, L. Van Vooren, and W. Bauwens (2000, July). Experience and organistation of automated measuring stations for river water quality monitoring. In Proceedings 1st IWA World Water Congress, Paris, France.

Vandenberghe, V., P. L. M. Goethals, A. Van Griensven, J. Meirlaen, N. De Pauw, P. Vanrolleghem, and W. Bauwens (2005). Application of automated measurement stations for continuous water quality monitoring of the dender river in flanders, belgium. *Environmental Monitoring and Assessment 108*(1-3), 85–98.

WSDOT (2008). *Water Quality Monitoring Databse User's Guide* (1 ed.). Washington State Department of Transportation.

Appendix A

Maintenance

Next figures show the set of tables used for maintenance.

Date	Starting time	Ending time	Comment
DD/MM/YYYY	hh:mm	hh:mm	Activities developed at the station's visit

Figure A.1: Table used to indicate all tasks developed at every visit to the stations.

		CONTROL CHART				
		Temperature - °C				
Date	Time	Conductivity	pH	LDO	Ammo:	Thermometer
DD/MM/YYYY	hh:mm					

Figure A.2: Table used for the temperature's control chart. Measured values.

								Control Limits							
$\Delta T = T_{thermomètre} - T_{capteur}$				Standar Deviation				Conductivity		pH		LDO		Ammolyser	
Conductivity	pH	LDO	Ammo:	Conductivity	pH	LDO	Ammolyser	Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A.3: Table used for the temperature's control chart. Calculation of the limits.

CLEANING/CALIBRATION/CONTROL CHART								
Before	After	% difference	Date	Starting time	Ending time	Comment	Slope	Offset calib

Figure A.4: Table used for the conductivity's control chart. Measured values.

				Control Limits	
CONTROL CHART			Standar Deviation	Conductivity	
Std sol.(uS/cm)	Value with std. sol. (uS/cm)	(Std sol)-(value with std sol.) (uS/cm)	Cond	Lower	Upper
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0

Figure A.5: Table used for the conductivity's control chart. Calculation of the limits.

CLEANING/CALIBRATION / CONTROL CHART									
Value Before	Value After	% difference	Date	Starting time	Ending time	Value at buffer1 (7)	Value at buffer2 (10)	Slope mVpH	Comment
			DD/MM/YYYY	hh:mm	hh:mm				

Figure A.6: Table used for the pH's control chart. Measured values.

CONTROL CHART						Control Limits			
(Std solution)- (Value at Std solution)		% difference		Standar Deviation		pH7		pH10	
ΔpH 7	ΔpH 10	pH7	pH10	pH7	pH10	Lower	Upper	Lower	Upper
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0
7	10			0	0	0	0	0	0

Figure A.7: Table used for the pH's control chart. Calculation of the limits.

CLEANING/CALIBRATION / CONTROL CHART								
Before	After	% difference	Date	Starting time	Ending time	Slope	Value in the bag	Comment
			DD/MM/YYYY	hh:mm	hh:mm			

Figure A.8: Table used for the LDO's control chart. Measured values.

CONTROL CHART		Standar Deviation	Control Limits	
			DO	
100% - value	% difference	DO	Lower	Upper
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0
100		0	0	0

Figure A.9: Table used for the LDO's control chart. Calculation of the limits.

CLEANING/CALIBRATION/CONTROL CHART									
Before (NTU)	After (NTU)	% difference	Date	Starting time	Ending time	Value with nano water	Value with Formazine 200 NTU	Value with Formazine 800 NTU	Comment

Figure A.10: Table used for the Solitax's control chart. Measured values.

						Control Limits					
			Standar Deviation			nano water		200 NTU		800 NTU	
Δ nano water	Δ 200 NTU	Δ 800 NTU	Nano water	200 NTU	800 NTU	Lower	Upper	Lower	Upper	Lower	Upper
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0
0	200	800	0	0	0	0	0	0	0	0	0

Figure A.11: Table used for the Solitax's control chart. Calculation of the limits.

CLEANING/CALIBRATION											
Temp		Ph		K		NH4		Date	Starting time	Ending time	Comment
Before	After	Before	After	Before	After	Before	After				

Figure A.12: Table used for the ammo::lyser's control chart. Measured values.

Control Charts						Control Limits			
Lab results				Standar Deviation		Δ pH		Δ Conc K+	
pH sensor	K+	Δ pH	Δ Conc K+	Δ pH	Δ Conc K+	Lower	Upper	Lower	Upper
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

Figure A.13: Table used for the ammo::lyser's control chart. Calculation of the limits.

CLEANING/CALIBRATION															
NO3 (mg/L NO3-N)			TurbiditY(FTU)			TOC (mg/L)			DOC (mg/L)			Date	Starting time	Ending time	Comment
Before	After	% difference	Before	After	% difference	Before	After	% difference	Before	After	% difference				

Figure A.14: Table used for the spectro::lyser's control chart. Measured values.

Control Charts								
Lab results			Standar Deviation		Control Limits			
					Before		After	
NO3-	Δ Conc Before cleaning	Δ Conc After cleaning	Before	After	Lower	Upper	Lower	Upper
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0
	0,00	0,00	0	0	0	0	0	0

Figure A.15: Table used for the spectro::lyser's control chart. Calculation of the limits.

			Lab values					
Ana::pro ID	Date		MeS (mg/L)	NO3-N (mg/L)	NH4-N (mg/L)	K (mg/L)	TOC (mg/L)	COD (mg/L)
ID Number	DD/MM/YYYY hh:ss	ammolyser						
		spectrolyser						

Figure A.16: Table used for the lab results. Measured values into the lab.

online measurement						
MeS (mg/L)	NO3-N (mg/L)	NH4-N (mg/L)	K (ppm)	TOC (mg/L)	DOC (mg/L)	Comment

Figure A.17: Table used for the lab results. Measured values by the sensors.

Appendix B

Database user's guide

This *User Guide* is intended to describe the model*EAU* Database, dat*EAU*base, for users who have no previous experience with it. More specifically, this guide should give the reader a basic understanding of the following concepts:

- What the dat*EAU*base is
- How to introduce information to the database
- How to get information from the database
- How to maintain the database

The information needed to analyse and use the database is provided, including clear definition of each data elements as well as its data collection frequency and procedure. To ensure the integrity of the database, only those data elements that have undergone a minimum level of quality control and assurance checks will be made available for release to the public.

To sum up, the database has been compiled in good faith with the intent of assisting user in the design, analysis, and management of measured environmental parameters.

B.1 Introduction

model*EAU* is a research team built around the Canada Research Chair on Water Quality Modelling that is held by Peter A. Vanrolleghem since February 2006. The research focus is mainly methodological, looking data collection, data quality evaluation, development of new models and improved modelling approaches, model-based optimization and supporting software.

Inside the water quality modelling, the tasks of storage, analysing and interpretation of the collected data take an important role. To facilitate and homogenize this work a database has been developed. It can serve as a tool to inform about environmental parameters in the model*EAU* group.

B.2 Database structure and content

Dat*EAU*base is an Access database accompanied by user-friendly tools to introduce and query data from any kind of environmental parameter.

The purpose of the database is store information in a useful way. It is comprised of multiple tables that contain records and fields. The fields describe the type of information stored, and records are the items in the database.

In order to provide the needed flexibility in the database model, a two-table design is included for organization and monitoring data: primary tables and lookup tables. The second ones are required to define in detail the codes contained in the primary tables.

The primary tables included in dat*EAU*base are WaterQuality, Hydraulics and Weather. The main structure of the primary tables was created as general as possible for any environmental parameter. On the other hand, the lookup tables are behind primary tables defining each element on them and keeping data integrity in the database environment.

The relational diagram for Water Quality data shown in figure B.1 contains descriptions of the primary tables as well as the numerous lookup tables required to define in detail the codes contained in the primary tables. For Hydraulics and Weather relational diagrams are analogous to the Water Quality relationship structure.

B.3 Database dictionary

The database dictionary is a comprehensive synopsis of the model*EAU* database, intended to facilitate the understanding and general use of the database by the user. This database dictionary is divided according to:

- Database table definitions
- Glossary of data elements

Database Table Definitions

Another important aspect of the database is the description parameters. The Database Table Definitions provide an overview of the dat*EAU*base. It lists and describes each table in the database, according to the modules presented in the section B.2.

The main structure of the primary tables designed for water quality, hydraulics and weather parameters designed it is shown in the table B.1.

Each characteristic element contained on the primary table has a related lookup table. These lookup tables permit to specify and determine every identification used on the

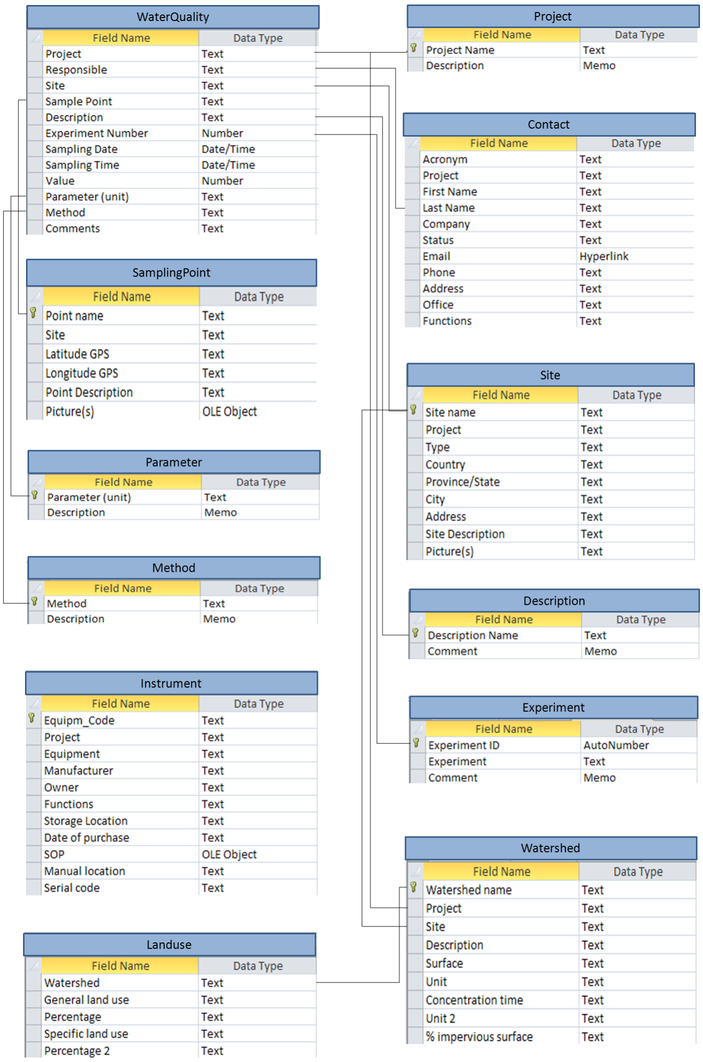


Figure B.1: Relationships diagram for organization and monitoring program elements of the dat*EAU*base.

primary tables.

With the relationships between the lookup tables and the primary data tables and enforcing referential integrity, data managers are restricted to entering only valid lookup table values into the primary data tables.

The lookup tables designed for dat*EAU*base are:

Table B.1: Data fields contained on primary tables.

Field	**Description**	**Format**
Project	Project identifier name	Text
Responsible	Last name identifying the person who managed the data	Text
Site	Situation of the sampling area	Text
Sample Point	Position on the site where readings were taken	Text
Description	Experiment type description	Text
Experiment Number	Number of replicas for an experiment	Number
Sampling Date	Date on which the WQ, hydraulics and weather readings were taken	Date/Time (dd/mm/yyyy)
Sampling Time	Time at which the WQ, hydraulics and weather readings were taken	Date/Time (hh:mm:ss)
Value	Parameter value	Number
Parameter (unit)	Code identifying parameter name and the abbreviation of the units of the parameter value	Text
Method	Code identifying field/laboratory test procedure	Text
Comments	Comments related to sampled parameter value	Text

1. **Contacts lookup table**: Information about all people who is involved in any project (Table B.2).

Table B.2: Data contained on contact lookup table.

Field	**Description**	**Format**
Acronym	Initials of the First and Last name	Text
Project	Name of the project	Text
First Name	First Name of who is involved in a project	Text
Last Name	Last Name of who is involved in a project	Text
Company	Name of the company who is working on it	Text
Status	Position inside the company	Text
E-mail	E-mail address of the contact	Hyperlink
Phone	Phone number of the contact	Text
Address	Address of the company	Text
Office	Office number	Text
Functions	Which functions the contact has	Text

2. **Description lookup table**: Details about the experiment type used on any

project (Table B.3).

Table B.3: Data contained on description lookup table.

Field	Description	Format
Description Name	Experiment type description	Text
Comment	Comments related to the description field	Memo

3. **Experiments lookup table**: Specifies all experiments used in any project and their identification (Table B.4).

Table B.4: Data contained on experiment lookup table.

Field	Description	Format
Experiment ID	Experiment identification	Text
Experiment	Name of the experiment	Text
Comment	Comments related to the experiment	Memo

4. **Instruments lookup table**: Characterizes the instruments used in any project with their identification code (Table B.5).

Table B.5: Data contained on instrument lookup table.

Field	Description	Format
Equipm_Code	The unique code of the instrument	Text
Project	Project identifier name	Text
Equipment	Name of the instrument used in a project	Text
Manufacturer	Name of the equipment manufacturer	Text
Owner	Owner of the equipment	Text
Functions	Functions of the instrument	Text
Storage Location	Place where the instrument is stored	Text
Date of Purchase	Date and place when and where the instrument was bought	Text
SOP	Standard operation procedure to use the instrument	OLE Object
Manual Location	Place where the manual is stored	Text
Serial Code	Serial code identification of the instrument	Text

5. **Land uses lookup table**: Details the land use where the sampling points are situated (Table B.6).

6. **Methods lookup table**: Detail of the methods employed in any project (Table B.7).

Table B.6: Data contained on land use lookup table.

Field	Description	Format
Watershed	Name of the watershed	Text
General Land Use	General land use of the watershed	Text
Percentage	Percentage of the general land use of the watershed	Text
Specific Land Use	Specific land use of the watershed	Text
Percentage 2	Percentage of the specific land use of the watershed	Text

Table B.7: Data contained on method lookup table.

Field	Description	Format
Method	Method code identifying field/laboratory test procedure	Text
Description	General description of the method	Memo

Table B.8: Data contained on parameter lookup table.

Field	Description	Format
Parameter (units)	Parameter code identifying units with its abbreviation	Text
Description	General description of the parameter	Memo

7. **Parameters lookup table**: Describes all measured parameters in any project (Table B.8).

8. **Projects lookup table**: Brief of all developed projects (Table B.9).

Table B.9: Data contained on project lookup table.

Field	Description	Format
Project Name	Project code identifying	Text
Description	General description of the project	Memo

9. **Sampling Points lookup table**: Exact situation of the sampling points of any project (Table B.10).

10. **Sites lookup table**: Describes the site that any project takes place (Table B.9).

11. **Watersheds lookup table**: Details the watershed places where any parameter is measured (Table B.12).

Table B.10: Data contained on sampling point lookup table.

Field	Description	Format
Point Name	Sampling point located on a site	Text
Site	Site name	Text
Latitude GPS	Coordinates GPS - Latitude	Text
Longitude GPS	Coordinates GPS - Longitude	Text
Point description	General description of the sampling point	Text
Picture(s)	Pictures file(s)	OLE Object

Table B.11: Data contained on site lookup table.

Field	Description	Format
Site Name	Name of the site	Text
Project	Name of the project	Text
Type	Type of sampling site	Text
Country	Location Country	Text
Province/State	Location Province/State	Text
City	Location City	Text
Address	Address location	Text
Site Description	General description of the site	Text
Picture(s)	Pictures of the site	OLE Object

Table B.12: Data contained on watershed lookup table.

Field	Description	Format
Watershed Name	Name of the watershed	Text
Project	Name of the project	Text
Site	Name of the site	Text
Description	Short description of the watershed	Text
Surface	Surface of the watershed	Text
Unit	Abbreviation of the units of surface value	Text
Concentration Time	Time of concentration of the watershed	Text
Unit 2	Abbreviation of the units of concentration time value	Text
% Impervious Surface	Percentage of the impervious surface	Text

Glossary of Data Elements

The glossary of the data elements provides a brief definition of each data element, by table. It should be a useful reference to users when formulating a data request and also as a reference source during future analyses of the data.

For each table in the database, it provides a definition of all fields residing in these tables, including:

- **Field**: The field names were established so as to provide a general understanding of the data element.
- **Description**: Provides a definition of the data element.
- **Format**: Whether the data element is a *description, number, Date/time, Hyperlink, Memo, OLE Object.*

The glossary of the data elements will be used as a second reference to the user for data introduction, formulating a data request, and also as a future reference when performing analyses of the data.

Although the data glossary is a good reference source to verify the definition of each data element, a more comprehensive description of the data collection methods employed in the different experiments can be found in the respective SOP.

B.4 Database update

As the work developed for model*EAU* is a field in continuous development, every effort will be made to augment the database. The updates will not be regularly scheduled, so users have to contact with the person in charge of the database to communicate new fields or changes on their projects.

B.5 Data introduction

To facilitate the data introduction to the database, a user-friendly interface has been created with Microsoft Excel helped by Visual Basic for Applications (VBA) code (Figure B.2).

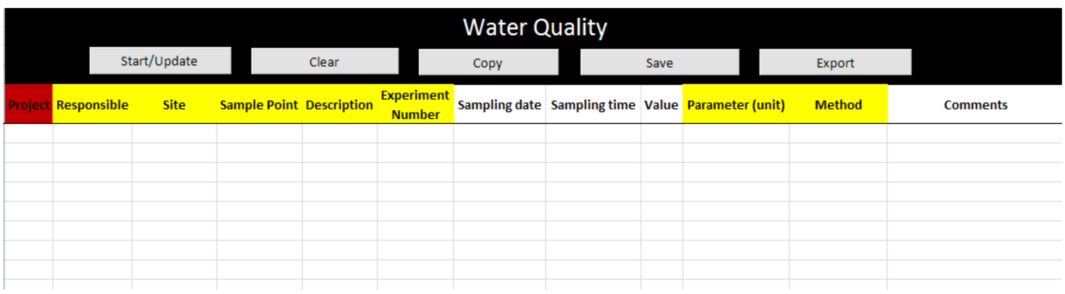


Figure B.2: Microsoft Excel interface to import data to the database.

This tool has been developed to facilitate the data importation from the user without the necessity to open the database and work directly on it.

Sampling date, sampling time and value are metadata and they must be filled in by hand. The other fields are the specific description of the measured value. They should be completed selecting one of the options from the corresponding pick list. The pick lists shown on the responsible, site, sample point, description, experiment number, parameter and method fields depend on the project selected.

Fields description

As it is observed on the figure B.2, several buttons and coloured cells can be perceived on the interface. The coloured cells permit to distinguish the different types of data on the interface. The colours meaning are:

- *Red*: Designates a description field that has to be filled first. Different options are shown on a pick list, one project must be chosen before one can fill the other description fields.
- *Yellow*: Specifies description fields. Each one has to be selected from the specific pick list. The elements of every pick list depend on the project.
- *White*: Indicates the metadata columns. These columns have to be filled by hand.

Moreover, each button at the top has a different function to facilitate the data management. The following functions are detailed.

- Start/Update button permits to start the selection from the pick lists. On the other hand, if there is any new element added in one of the pick lists, pressing this button will update the list.
- Clear button deletes any value on the table.
- Copy button copies the first line of description data until the last measured value.
- Save button saves the data table in a new excel file without the buttons and the VBA code.
- Export button exports the data to the database, after checking the metadata format and saves the file.

Data importation

To introduce data to the database, next steps are recommended to be followed.

1. Open the table interface file. As there are some macros on it, it is necessary to enable them before to start the data introduction.

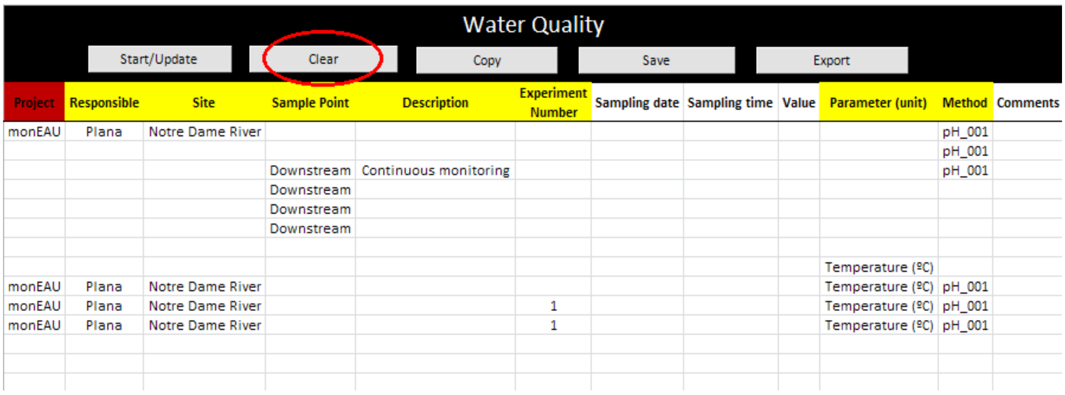


Figure B.3: Microsoft Excel interface to import data to the database before cleared out.

2. If there is any cell stored, press the Clear button to delete them before to start. In example, figure B.3.
3. Press the Start/Update button (Figure B.4).

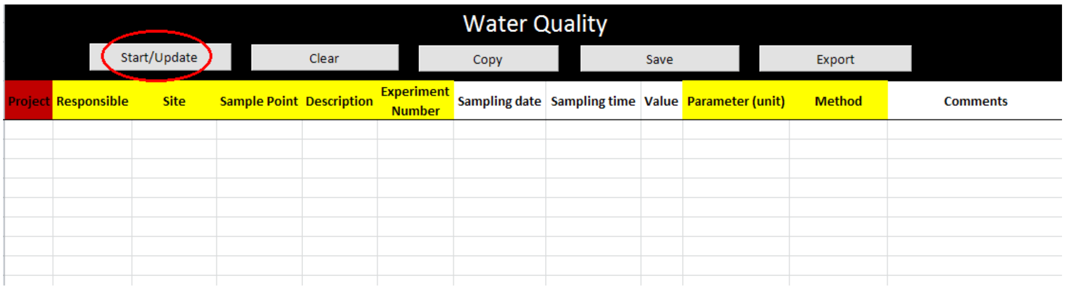


Figure B.4: Microsoft Excel interface to import data to the database.

4. Choose a project on the first row to the red column (Figure B.5).

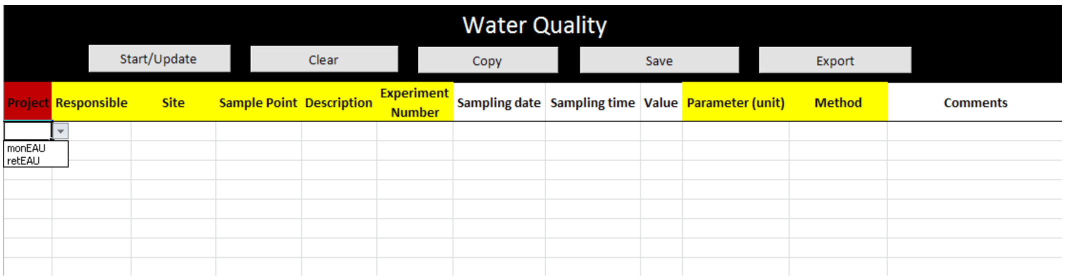


Figure B.5: Projects pick list.

5. Only the first row, complete the other description fields (Yellow columns) selecting an element on each pick list (Figure B.6).

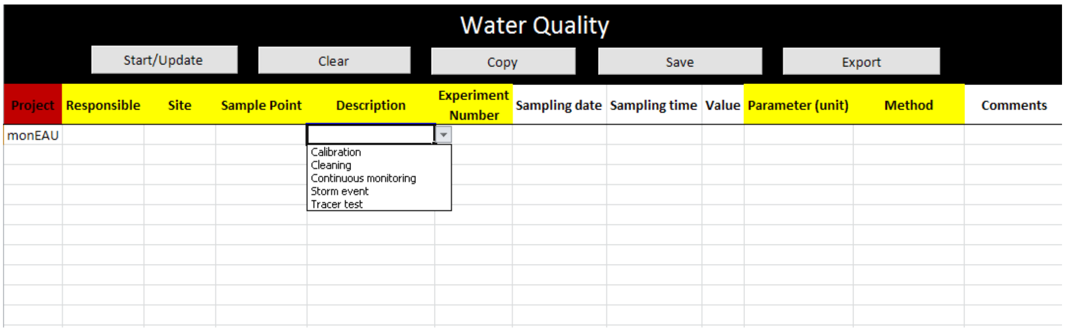


Figure B.6: Example of one pick list to fill the description fields.

On the Experiment Number column, if there is more than one measured value for the same sample like laboratory test, it is needed to fill the corresponding number on the first series. In example, figure B.7.

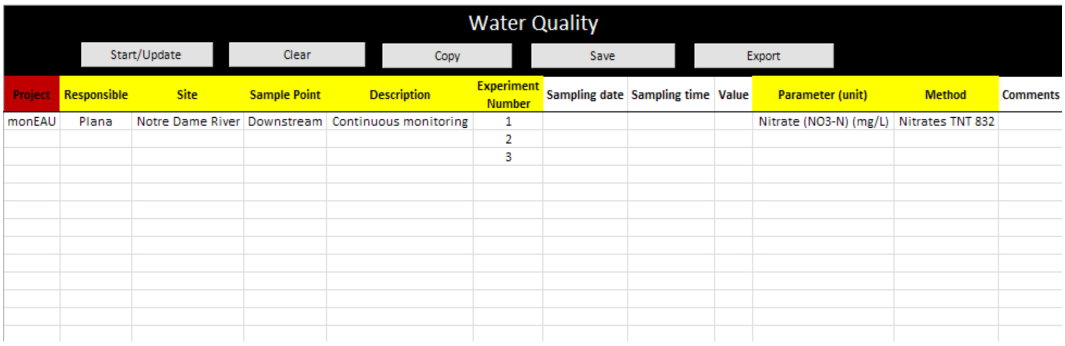


Figure B.7: Example of another possibility for the Experiment Number column.

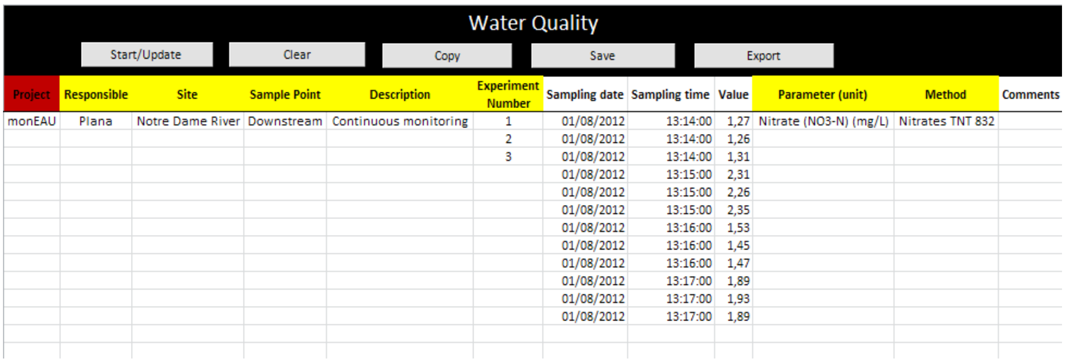
6. Insert the measured values (Figure B.8).

7. Copy the first row until last row by pressing the *Copy* button.

   In case of more than one Experiment Number at the same table, the series is going to be copied by blocks. In example, figure B.9.

8. Save the data on a table by the *Save* button (Figure B.10).

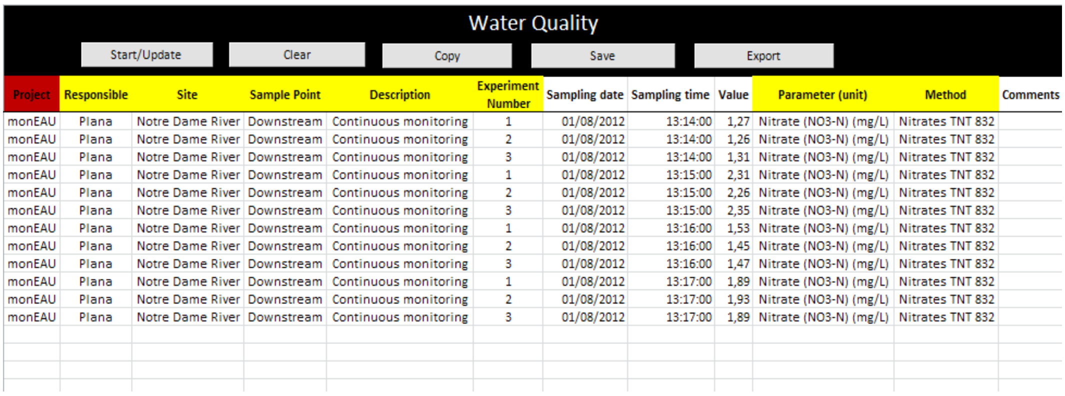
   It is possible to save the data table with the same name several times.


Water Quality

Start/Update | Clear | Copy | Save | Export

Project	Responsible	Site	Sample Point	Description	Experiment Number	Sampling date	Sampling time	Value	Parameter (unit)	Method	Comments
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	13:14:00	1,27	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
					2	01/08/2012	13:14:00	1,26			
					3	01/08/2012	13:14:00	1,31			
						01/08/2012	13:15:00	2,31			
						01/08/2012	13:15:00	2,26			
						01/08/2012	13:15:00	2,35			
						01/08/2012	13:16:00	1,53			
						01/08/2012	13:16:00	1,45			
						01/08/2012	13:16:00	1,47			
						01/08/2012	13:17:00	1,89			
						01/08/2012	13:17:00	1,93			
						01/08/2012	13:17:00	1,89			

Figure B.8: Measured values introduction example.


Water Quality

Start/Update | Clear | Copy | Save | Export

Project	Responsible	Site	Sample Point	Description	Experiment Number	Sampling date	Sampling time	Value	Parameter (unit)	Method	Comments
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	13:14:00	1,27	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	2	01/08/2012	13:14:00	1,26	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	3	01/08/2012	13:14:00	1,31	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	13:15:00	2,31	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	2	01/08/2012	13:15:00	2,26	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	3	01/08/2012	13:15:00	2,35	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	13:16:00	1,53	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	2	01/08/2012	13:16:00	1,45	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	3	01/08/2012	13:16:00	1,47	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	1	01/08/2012	13:17:00	1,89	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	2	01/08/2012	13:17:00	1,93	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	
monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring	3	01/08/2012	13:17:00	1,89	Nitrate (NO3-N) (mg/L)	Nitrates TNT 832	

Figure B.9: Copy example for more than one experiment number at the same table.

9. Import the data to the database with the *Import* button.

   Before to complete the data importation, the function of the *Import* button has some steps before included:

   a) Check if the Sampling Date, Sampling Time and Value are in the correct format (Table B.13)

      If the date, time and value format is not correct a message will appear (Figure B.11).

   b) Data import confirmation (Figure B.12). If the import operation is aborted, the data is not introduced into the database.

   c) Save the data on a table. It is possible to save the table with the same name as previous times.

   d) Data importation to the dat*EAU*base (Figure B.13).

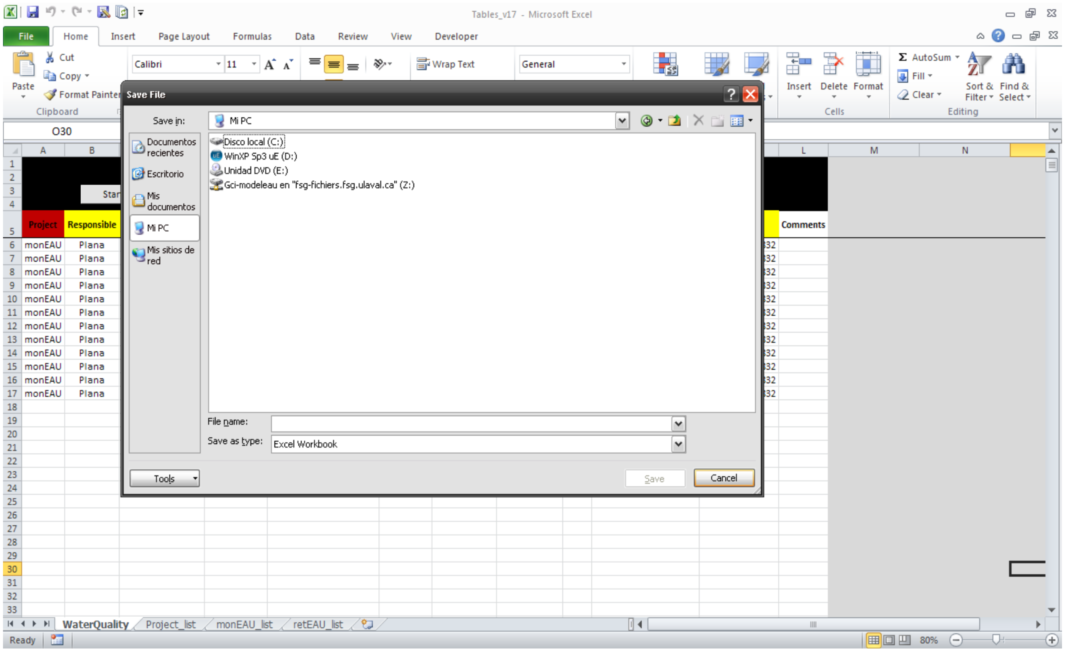


Figure B.10: *Save as* box.

Table B.13: Format of the meeasured data to be filled on the dat*EAU*base.

Field	Format
Sampling Date	DD/MM/YYYY
Sampling Time	hh:mm:ss
Value	Number

e) Clear the table.

Update importation interface

The updates can be introduced easily on the importation interface. Even though, users have to contact with the person in charge of the database to communicate new fields or changes on their projects before to modify it.

Adding a new element

To add a new element on a pick list, the next steps have to be followed.

- Go to the specific project sheet. In example, mon*EAU* project list (Figure B.14).
- Add the new element to the corresponding column.

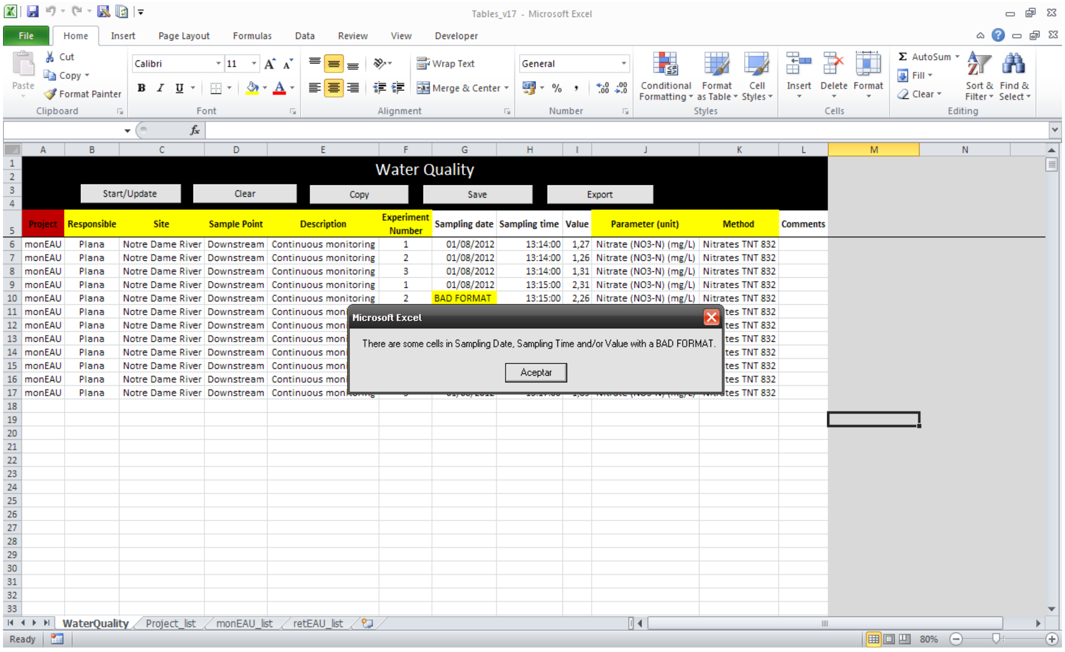


Figure B.11: Box to notify that there is data introduced with a wrong format.

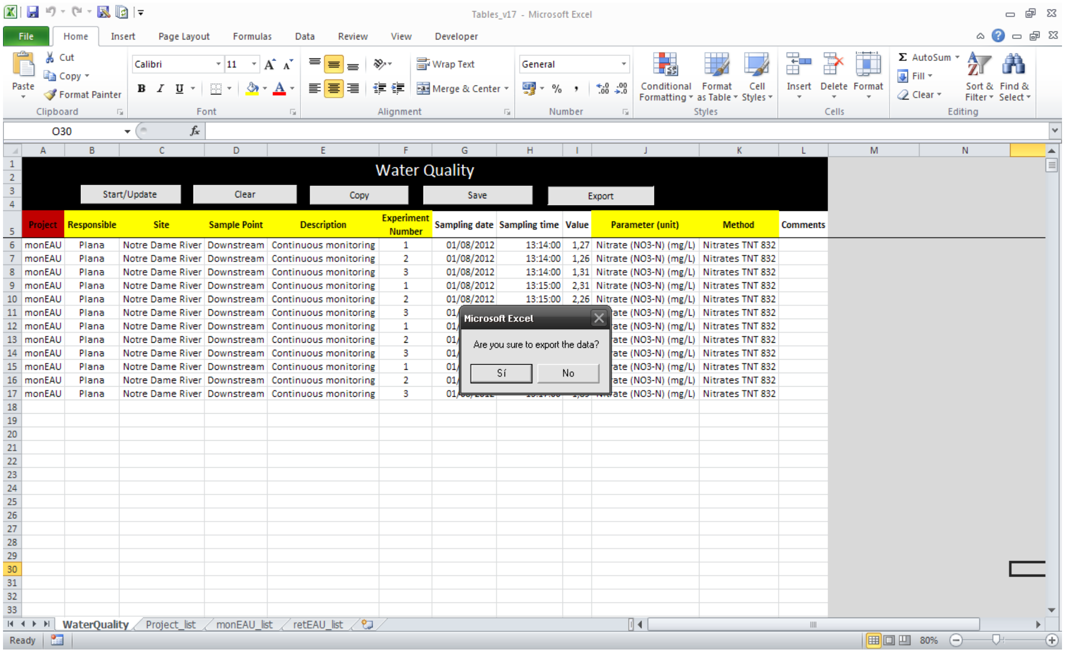


Figure B.12: Box to confirm the data importation.

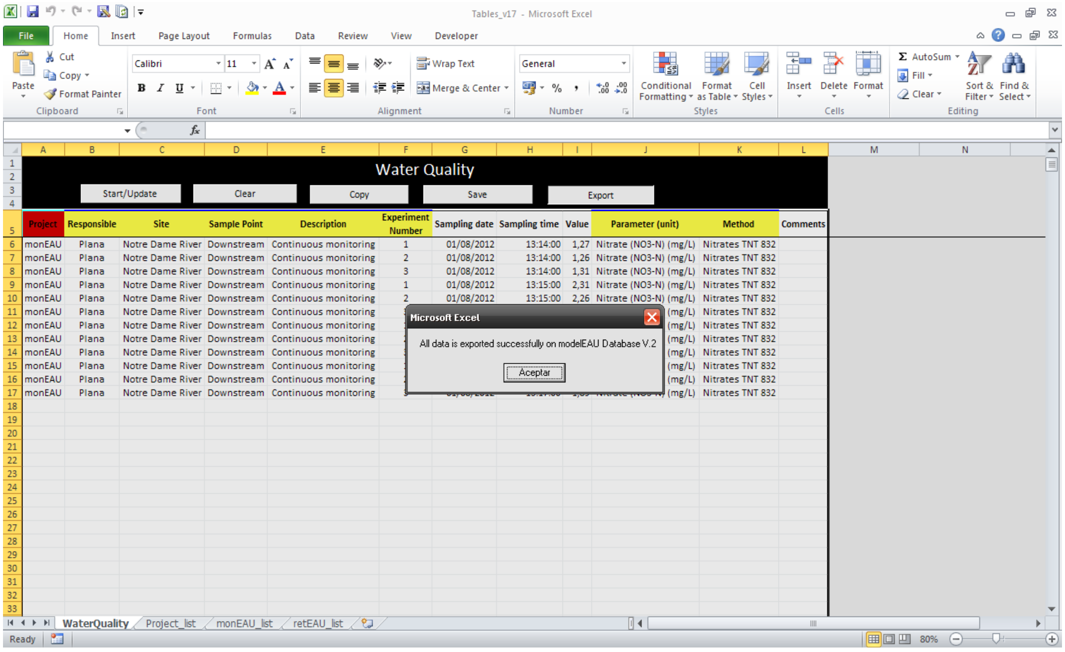


Figure B.13: Confirmation message when the importation action is done successfully.

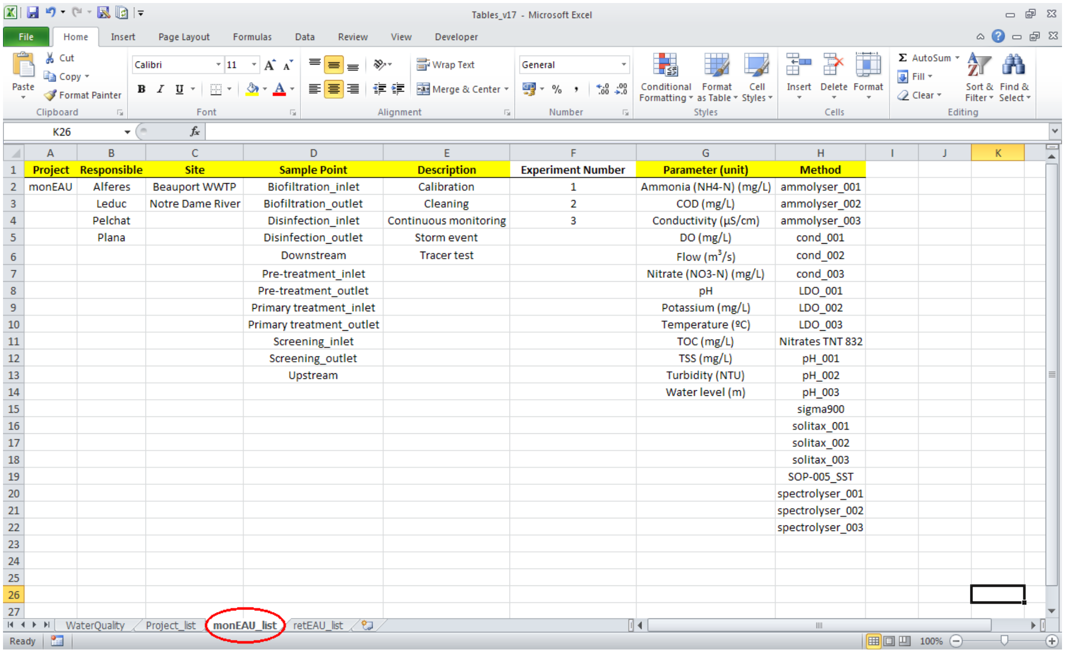


Figure B.14: Example of the pick lists table structure.

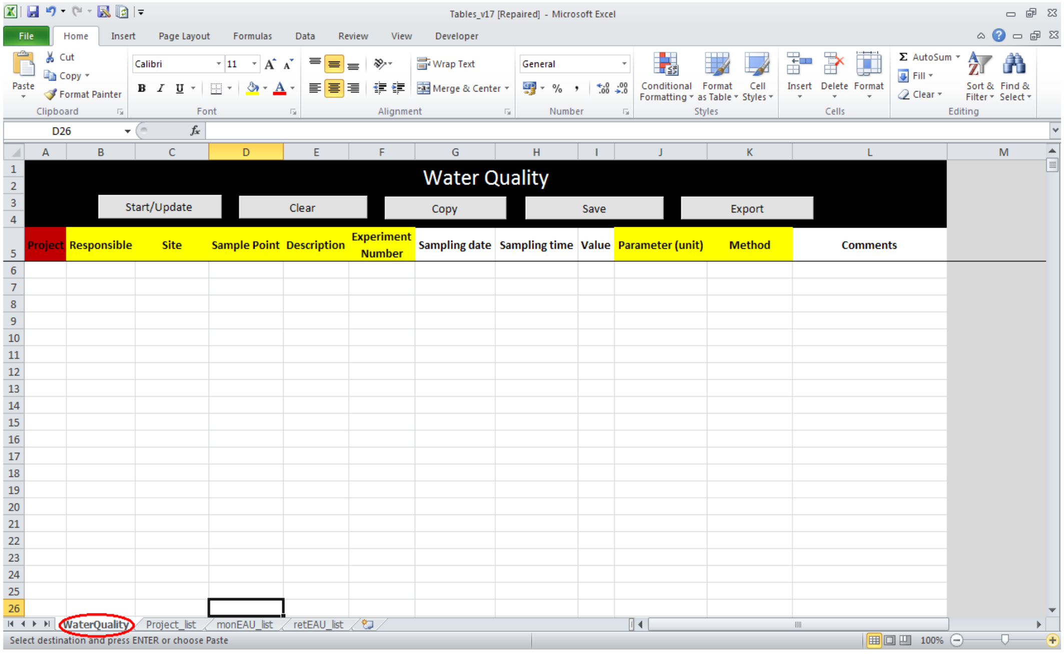


Figure B.15: Example of the pick lists table structure.

- Come back to the primary table. In example, WaterQuality sheet (Figure B.15).
- Press *Start/Update* button to begin the data introduction.

Adding a new project

Below, the procedure to add a new project is detail.

1. Go to the Project_list sheet (Figure B.16).
2. Add to the list the name of the new project.
3. Create a new sheet named projectname_list after the last created project sheet (Figure B.17).
4. Build a table with the same structure as figure B.14.
5. Come back to the primary table.
6. Press *Start/Update* button to begin the data introduction.

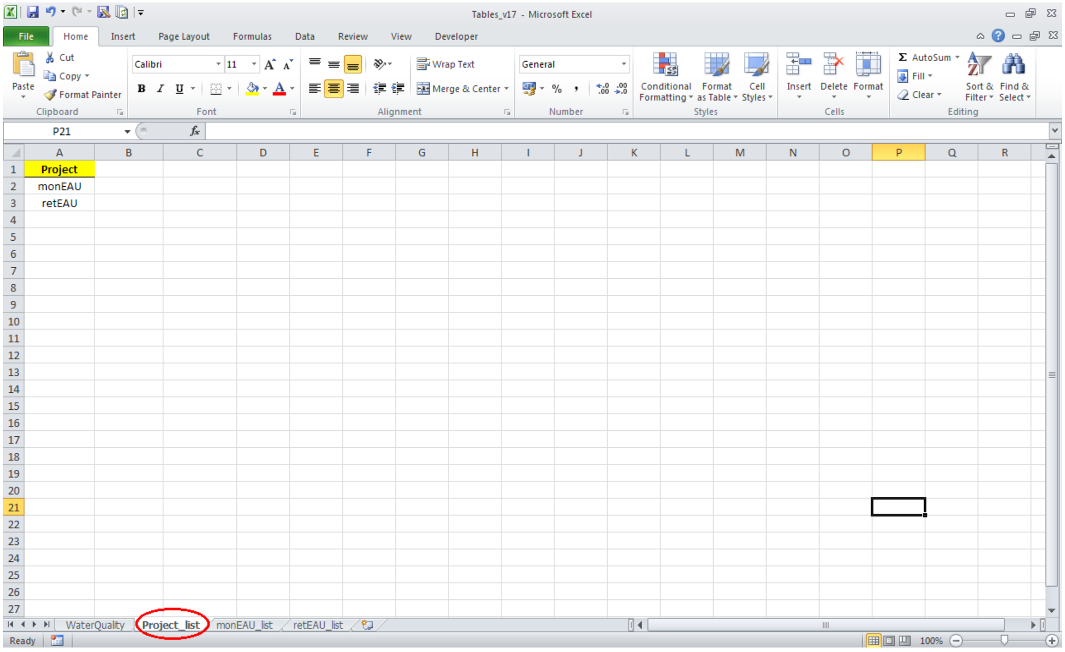


Figure B.16: Projects list sheet.

B.6 Requesting and plot data

The tool to export the data has been developed in R. It allows to open the database, query the data, create graphics and evaluate them by specific functions.

R is an open source software environment and language for statistical computing and graphics. It can be downloaded for free at the web page of R-project. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

R runs on Microsoft Windows platforms using Object Database Connectivity (ODBC) package. Providing a file name of a data base object or a Data Source Name (DSN) and other connection information, and the R software can then interact with the database. Access database files can then directly interact with R.

Packages

Furthermore, R can be extended via *packages*. This packages are available through the R distribution and the CRAN family of Internet sites covering a very wide range of modern statistics. In this case to run the created functions, two specific packages are required:

- **ROBC**: This package provides access to databases (including Microsoft Access

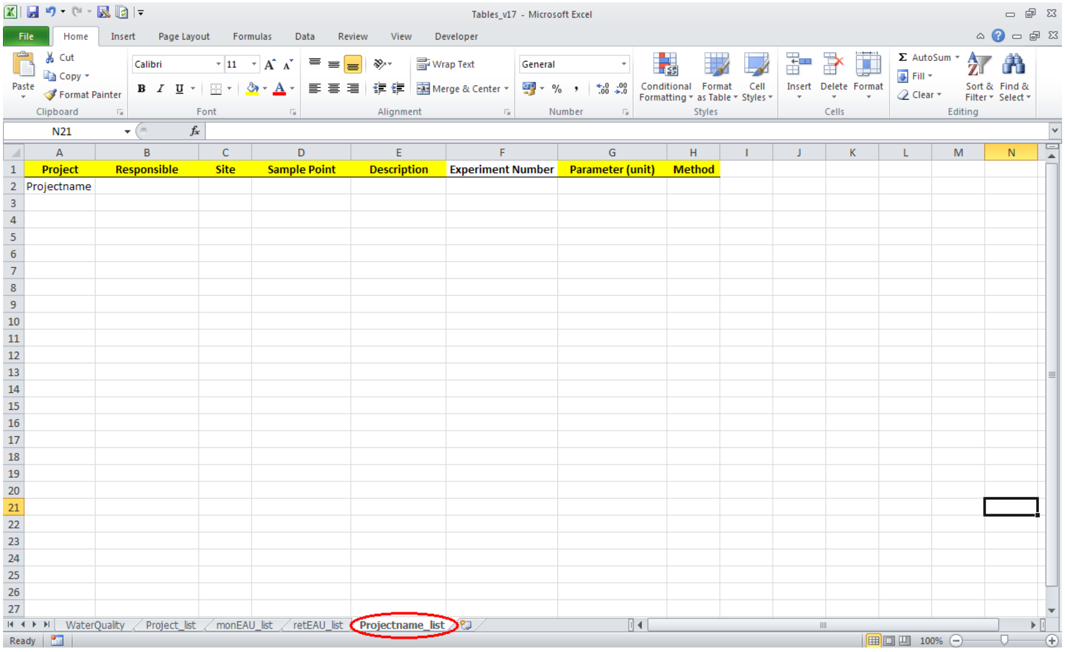


Figure B.17: New sheet creation for a new project.

and Microsoft SQL Server) through an ODBC interface.

- **gWidgetsRGtk2**: This package is port of gWidgets API to RGtk2. Th GTK toolkit is interfaced via the RGtk2 package, of which in turn is derived from RGtk package.

  The packages RGtk2, cairoDevice, gWidgets also are required. These can be installed by following the dependencies for gWidgetsRGtk2.

They have to be installed before running the created functions. On Windows, to install packages there are two possibilities:

1. Use the command:

```
install.packages("packagename")
```

2. Menu *Packages* choosing Install packages option and selecting the package desired.

Functions

As a user-friendly tools to request data from the database and make graphics to evaluate them, three main functions have been created:

- **Open the database function**: It permits to open a table from a database. It displays the opened data table.
- **Query data function**: It is a query data interface. The output is the data selected to be plotted.
- **Plot data function**: It allows to make scatterplots of up to four series of data.

To execute these functions, the steps below are recommended to be followed.

1. Define the directory.

   The command to set a directory in R is:

   ```
   setwd(dir)     # dir is a character string and it defines the directory
       path.
   ```

   On the other hand, another option is possible: know which is the actual directory set to put the functions and the data on it. The command to get an absolute filepath representing the current working directory in R is:

   ```
   getwd()
   ```

2. Read the function to open the database developed in Microsoft Access.

   ```
   source("opendb.R")
   ```

3. Execute the opening database function.

   ```
   opendb("file", "tab")
   ```

   file and *tab* are the inputs of this function. On the *file* input, the filename of the database has to be indicated and on the *tab* input, requires the name of the table from the database to be opened.

4. A table with the exported data is displayed (i.e. figure B.18).
5. Read the function *doit*. It is a subfunction used on the *query* function.

   ```
   source("doit.R")
   ```

6. Read the function to query the data to be plotted.

   ```
   source("query.R")
   ```

Data: var1

	Project	Responsible	Site	Sample_Point	Description
1	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
2	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
3	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
4	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
5	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
6	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
7	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
8	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
9	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
10	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
11	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
12	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
13	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
14	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
15	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
16	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
17	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
18	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
19	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
20	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
21	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
22	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
23	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring
24	monEAU	Plana	Notre Dame River	Downstream	Continuous monitoring

Figure B.18: Table displayed after exporting the data from the database.

7. To execute this function the following instruction has to be ordered:

```
query(Dat1, Dat2)
```

   Dat1 and *Dat2* are the inputs of this function. These inputs mean *Start Date* and *End Date* to delimit the period of data to be plotted. The format of the inputs must be "YYYY-MM-DD".

8. After this a window is displayed (Figure B.19), showing several pick lists depending on the data exported from the dat*EAU*base.

9. Choose an element from each pick list to select which parameter is going to be plotted (i.e. figure B.20).

10. Fulfil the *assignto* field to designate a name to the selected variable (Figure B.21).

    It is suggested to name the variables var1, var2, ... in numerical order depending on how many variables are going to be plotted. Moreover, after assigning the variable name, it is sufficient to only pressing *accept* button before to defining and denominating next variable.

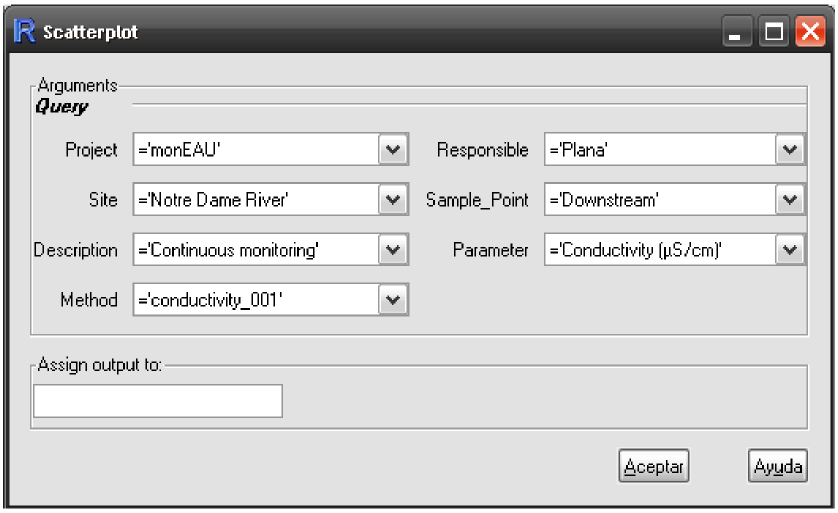


Figure B.19: Window displayed to query the data.

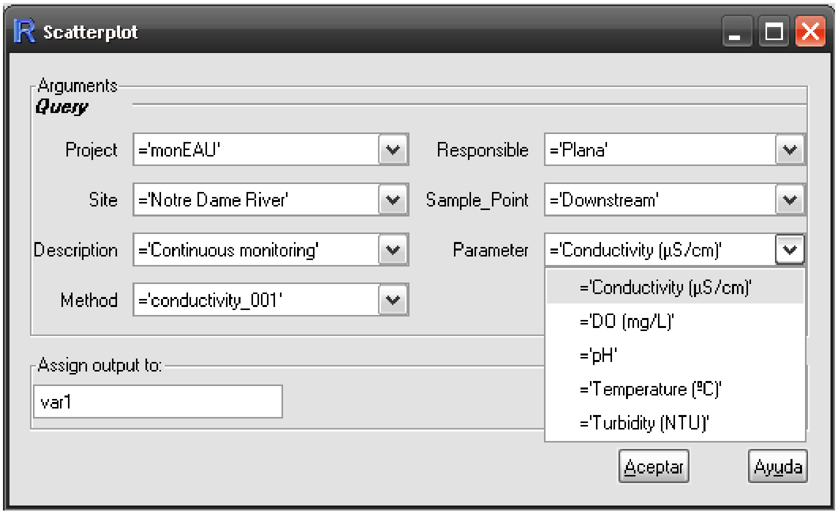


Figure B.20: Example of pick list to query the data.

11. Press *accept* button.
12. Read *plotgraph* function.

```
source("plotgraph.R")
```

13. Execute *plotgraph* function.

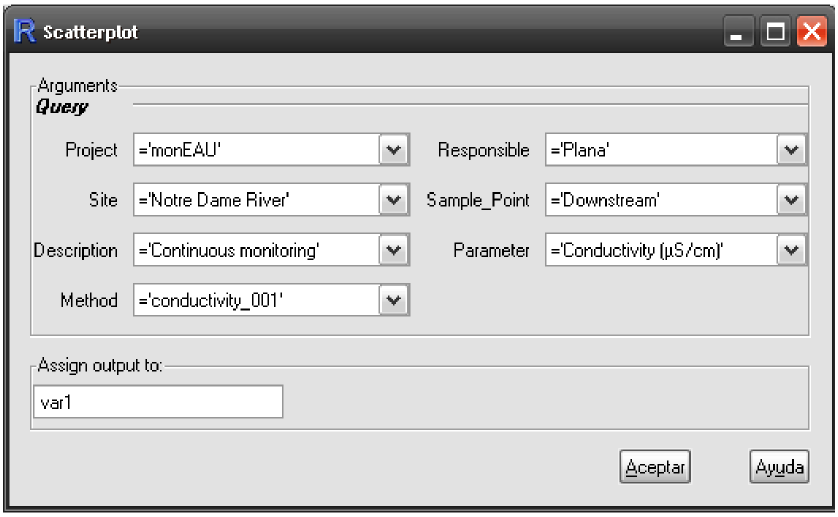


Figure B.21: Name designation to a queried parameter.

```
plotgraph(n,m)
```

Where n means the number of variables defined on the *query* function to be plotted in the same graphic. And m means how many y-axes want to be plot.

Only there is the possibility two axis graph when two parameters are going to be plotted.

14. A scatter plot is displayed (i.e. figure B.22).

    The plot is automatically saved as *graph.png* to the set directory. Before making another graphic, its name must be changed.

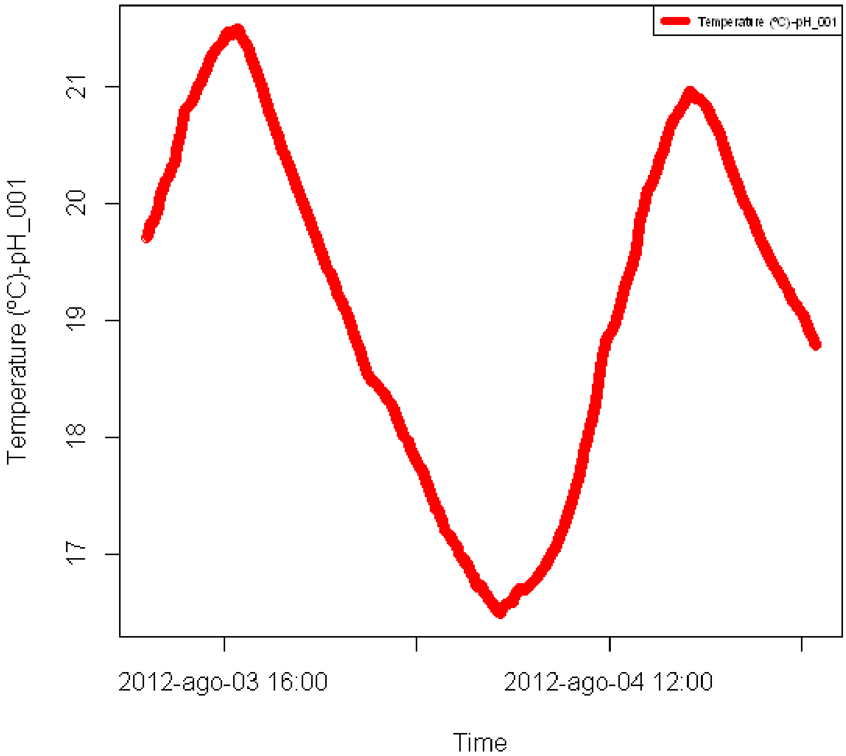


Figure B.22: Example of a scatterplot: Temperature graph measured by conductivity sensor at the downstream station.

Appendix C

User interface code

In this section the developed codes are presented.

C.1 Data importation interface code

VBA code for the Microsoft Excel user interface.

Dynamic pick lists code

```
Private Sub Worksheet_Change(ByVal Target As Range)

Dim LastRow As Long
Dim Responsible As String, Site As String, Sample As String, _
    Description As String, Experiment As String, Parameter As String, Method As
        String
Dim j As Long

Application.EnableEvents = False

' Select the sheet conciding with the cell "A2" value
For j = 3 To Sheets.count

    If StrComp(Sheets(j).Range("$A$2").Value2, Sheets(1).Range("$A$6").Value2,
        vbTextCompare) = 0 Then

        ' Find LastRow in Col B into the Sheet(j)
        LastRow = Sheets(j).Range("B" & Rows.count).End(xlUp).Row

        ' Select all Col B in Responsible
        For Each Value In Sheets(j).Range("B2:B" & LastRow)
            Responsible = Responsible & "," & Value
        Next Value

        'Sheet1.Range("B6").ClearContents:
        Sheet1.Range("B6:B100").Validation.Delete

        ' Create the Responsible Data Validation List
        With Sheet1.Range("B6:B100").Validation
```

```
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
       Operator:=xlBetween, Formula1:=Responsible
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

' Find LastRow in Col C into the Sheet(j)
LastRow = Sheets(j).Range("C" & Rows.count).End(xlUp).Row

' Select all Col C in Responsible
For Each Value In Sheets(j).Range("C2:C" & LastRow)
    Site = Site & "," & Value
Next Value

'Sheet1.Range("C6").ClearContents:
Sheet1.Range("C6:C100").Validation.Delete

' Create the Site Data Validation List
With Sheet1.Range("C6:C100").Validation
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
       Operator:=xlBetween, Formula1:=Site
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

' Find LastRow in Col D into the Sheet(j)
LastRow = Sheets(j).Range("D" & Rows.count).End(xlUp).Row

' Select all Col D in Responsible
For Each Value In Sheets(j).Range("D2:D" & LastRow)
    Sample = Sample & "," & Value
Next Value

'Sheet1.Range("D6").ClearContents:
Sheet1.Range("D6:D100").Validation.Delete

' Create the Sample Data Validation List
```

```
With Sheet1.Range("D6:D100").Validation
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
        Operator:=xlBetween, Formula1:=Sample
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

' Find LastRow in Col E into the Sheet(j)
LastRow = Sheets(j).Range("E" & Rows.count).End(xlUp).Row

' Select all Col E in Responsible
For Each Value In Sheets(j).Range("E2:E" & LastRow)
    Description = Description & "," & Value
Next Value

'Sheet1.Range("E6").ClearContents:
Sheet1.Range("E6:E100").Validation.Delete

' Create the Description Data Validation List
With Sheet1.Range("E6:E100").Validation
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
        Operator:=xlBetween, Formula1:=Description
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

' Find LastRow in Col F into the Sheet(j)
LastRow = Sheets(j).Range("F" & Rows.count).End(xlUp).Row

' Select all Col F in Responsible
For Each Value In Sheets(j).Range("F2:F" & LastRow)
    Experiment = Experiment & "," & Value
Next Value

'Sheet1.Range("J6").ClearContents:
Sheet1.Range("F6:F100").Validation.Delete
```

```
' Create the Parameter Data Validation List
With Sheet1.Range("F6:F100").Validation
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
        Operator:=xlBetween, Formula1:=Experiment
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

 ' Find LastRow in Col F into the Sheet(j)
LastRow = Sheets(j).Range("G" & Rows.count).End(xlUp).Row

' Select all Col F in Responsible
For Each Value In Sheets(j).Range("G2:G" & LastRow)
    Parameter = Parameter & "," & Value
Next Value

'Sheet1.Range("J6").ClearContents:
Sheet1.Range("J6:J100").Validation.Delete

' Create the Parameter Data Validation List
With Sheet1.Range("J6:J100").Validation
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
        Operator:=xlBetween, Formula1:=Parameter
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With

' Find LastRow in Col G into the Sheet(j)
LastRow = Sheets(j).Range("H" & Rows.count).End(xlUp).Row

' Select all Col G in Responsible
For Each Value In Sheets(j).Range("H2:H" & LastRow)
    Method = Method & "," & Value
Next Value

'Sheet1.Range("K6").ClearContents:
Sheet1.Range("K6:K100").Validation.Delete
```

```
        ' Create the Method Data Validation List
        With Sheet1.Range("K6:K100").Validation
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
                Operator:=xlBetween, Formula1:=Method
            .IgnoreBlank = True
            .InCellDropdown = True
            .InputTitle = ""
            .ErrorTitle = ""
            .InputMessage = ""
            .ErrorMessage = ""
            .ShowInput = True
            .ShowError = True
        End With
    End If
Next j

Application.EnableEvents = True

With Me.Cells
    .Columns.AutoFit
End With

End Sub
```

Start and update code

```
Sub Button2_Click()

' Start and update the data importation.

Dim LastRow As Long
Dim Project As String
Dim j As Long

Application.EnableEvents = False

' Find LastRow in Col A into the Sheet2
LastRow = Sheet2.Range("A" & Rows.count).End(xlUp).Row

' Select all Col A in Project
For Each Value In Sheet2.Range("A2:A" & LastRow)
    Project = Project & "," & Value
Next Value

' Sheet1.Range("A6").ClearContents:
Sheet1.Range("A6:A100").Validation.Delete
```

```
' Create the Data Validation List
   With Range("A6:A100").Validation
       .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,
          Operator:=xlBetween, Formula1:=Project
       .IgnoreBlank = True
       .InCellDropdown = True
       .InputTitle = ""
       .ErrorTitle = ""
       .InputMessage = ""
       .ErrorMessage = ""
       .ShowInput = True
       .ShowError = True
   End With

' Create lists ordered alphabetically

' Order alphabetically Project List
Sheet2.Columns("A:A").Sort Key1:=Sheet2.Range("A2"), Order1:=xlAscending,
    Header:=xlYes

' Order alphabetically the other lists on each project
For j = 3 To Sheets.count
   Sheets(j).Columns("B:B").Sort Key1:=Sheets(j).Range("B2"),
       Order1:=xlAscending, Header:=xlYes
   Sheets(j).Columns("C:C").Sort Key1:=Sheets(j).Range("C2"),
       Order1:=xlAscending, Header:=xlYes
   Sheets(j).Columns("D:D").Sort Key1:=Sheets(j).Range("D2"),
       Order1:=xlAscending, Header:=xlYes
   Sheets(j).Columns("E:E").Sort Key1:=Sheets(j).Range("E2"),
       Order1:=xlAscending, Header:=xlYes
   Sheets(j).Columns("F:F").Sort Key1:=Sheets(j).Range("F2"),
       Order1:=xlAscending, Header:=xlYes
   Sheets(j).Columns("G:G").Sort Key1:=Sheets(j).Range("G2"),
       Order1:=xlAscending, Header:=xlYes
Next j

Application.EnableEvents = True

End Sub
```

Clear the table code

```
Sub Button4_Click()

' Clear the table
```

```
Dim Data As Range

' Clear the range
Sheet1.Range("A6:L65536").ClearContents

End Sub
```

Copy and paste code

```
Sub Button10_Click()

' Copy and paste the first row

Dim LastRow1 As Long, LastRow2 As Long, LastRow3 As Long, LastRow5 As Long,
    LastRow6 As Long, LastRow7 As Long
Dim rngPaste As Range

' Find LastRow in Col G into the Sheet1
LastRow5 = Sheet1.Range("G" & Rows.count).End(xlUp).Row

' Find LastRow in Col H into the Sheet1
LastRow6 = Sheet1.Range("H" & Rows.count).End(xlUp).Row

' Find LastRow in Col I into the Sheet1
LastRow7 = Sheet1.Range("I" & Rows.count).End(xlUp).Row

If LastRow5 = LastRow6 Then
    If LastRow5 = LastRow7 Then

If IsEmpty(Sheet1.Range("A6")) Then
    MsgBox "You have to select a Project"
Else
    If IsEmpty(Sheet1.Range("B6")) Then
        MsgBox "You have to select a Responsible"
    Else
        If IsEmpty(Sheet1.Range("C6")) Then
            MsgBox "You have to select a Site"
        Else
            If IsEmpty(Sheet1.Range("D6")) Then
                MsgBox "You have to select a Sample Point"
            Else
                If IsEmpty(Sheet1.Range("E6")) Then
                    MsgBox "You have to select a Description"
                Else
                    If IsEmpty(Sheet1.Range("F6")) Then
                        MsgBox "You have to select a Experiment Number"
                    Else
```

```
If IsEmpty(Sheet1.Range("G6")) Then
    MsgBox "You have to select a Sampling Date"
Else
    If IsEmpty(Sheet1.Range("H6")) Then
        MsgBox "You have to select a Sampling Time"
    Else
        If IsEmpty(Sheet1.Range("I6")) Then
            MsgBox "You have to select a Value"
        Else
            If IsEmpty(Sheet1.Range("J6")) Then
                MsgBox "You have to select a Parameter"
            Else
                If IsEmpty(Sheet1.Range("K6")) Then
                    MsgBox "You have to select a Method"
                Else
                    ' Find LastRow in Col I into the Sheet1
                    LastRow1 = Sheet1.Range("I" &
                        Rows.count).End(xlUp).Row

                    ' Find LastRow in Col F into the Sheet1
                    LastRow2 = Sheet1.Range("F" &
                        Rows.count).End(xlUp).Row
                    LastRow3 = Sheet1.Range("F" &
                        Rows.count).End(xlUp).Row

                    ' Data to copy (First columns)
                    Set rngPaste = Sheet1.Range("A6:E6")
                    rngPaste.Copy

                    ' Paste
                    For i = 7 To LastRow1
                        Sheet1.Cells(i, 1).Select
                        ActiveSheet.Paste
                    Next i

                    ' Data to copy (Last 2 columns)
                    Set rngPaste = Sheet1.Range("J6:K6")
                    rngPaste.Copy

                    ' Paste
                    For i = 7 To LastRow1
                        Sheet1.Cells(i, 10).Select
                        ActiveSheet.Paste
                    Next i

                    ' Data to copy (Column F)
                    Set rngPaste = Sheet1.Range("F6:F" &
                        LastRow2)
                    rngPaste.Copy
```

```
                                                  ' Paste
                                                  Do Until LastRow1 = LastRow3
                                                      i = LastRow3 + 1
                                                      Sheet1.Cells(i, 6).Select
                                                      ActiveSheet.Paste
                                                      ' Find LastRow in Col F into the
                                                           Sheet1
                                                      LastRow3 = Sheet1.Range("F" &
                                                           Rows.count).End(xlUp).Row
                                                  Loop
                                              End If
                                          End If
                                      End If
                                  End If
                              End If
                          End If
                      End If
                  End If
              End If
          End If
      End If

      Else
          MsgBox "There is no the same number of data in Sampling Date, Sampling Time
               and Value"
      End If

End If

End Sub
```

Save the table code

```
Sub Button11_Click()

' Save the table with data.

Dim wb As String
Dim NewWbk As String
Dim NewShtName As Variant
Dim filepath As String
Dim fd As FileDialog

    ' Assign active workbook to variable wb
    wb = ActiveWorkbook.Name
```

```
' Create a FileDialog object as a Save As dialog box.
Set fd = Application.FileDialog(msoFileDialogSaveAs)

' Save As window
With fd
    .AllowMultiSelect = False
    .Title = "Save File"
    .Show
    If .SelectedItems.count > 0 Then
        filepath = .SelectedItems(1) & "\"
        NewShtName = .SelectedItems(1)
    Else
        Exit Sub
    End If
End With

Set fd = Nothing

' Add a new workbook
Application.Workbooks.Add
Range("A1").Select

' Save the new workbook and specify location and name
ActiveWorkbook.SaveAs Filename:=NewShtName & ".xlsx"

' Assign the active workbook to the NewWbk variable
NewWbk = ActiveWorkbook.Name

' Activate the initial window that the macro was run from, and activate the
    Form sheet
Workbooks(wb).Sheets("WaterQuality").Activate

'Select the range to copy
Sheet1.Range("A5:L65536").Select

' Copy the selected range
Selection.Copy

' Activate the new workbbook selecting sheet1
Workbooks(NewWbk).Sheets("Sheet1").Activate

' Paste it into the new workbook
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False

' Rename Sheet1
ActiveSheet.Name = "WaterQuality"

' Save and close New Workbook
```

```
    ActiveWorkbook.Save
    ActiveWorkbook.Close

    MsgBox "You can find the file in " & filepath

End Sub
```

Import the data code

```
Sub Button14_Click()

' Exports data from the active worksheet to a table in an Access database

Dim cn As ADODB.Connection
Dim r As Long, i As Long, count As Long, Clr As Long
Dim LastRow5 As Long, LastRow6 As Long, LastRow7 As Long
Dim Data As Range, cell As Range, col As Range, Cll As Range, RngColor As
    Range, Rng As Range
Dim msg As String
Dim CountColor As Integer, iVal As Integer

' Find LastRow in Col G into the Sheet1
LastRow5 = Sheet1.Range("G" & Rows.count).End(xlUp).Row

' Find LastRow in Col H into the Sheet1
LastRow6 = Sheet1.Range("H" & Rows.count).End(xlUp).Row

' Find LastRow in Col I into the Sheet1
LastRow7 = Sheet1.Range("I" & Rows.count).End(xlUp).Row

For Each cell In Range("G6:I" & LastRow7)
    cell.Interior.ColorIndex = 0 ' Color the cell White
Next cell

If LastRow5 = LastRow6 Then
    If LastRow5 = LastRow7 Then

' Check if there is any empty cell before to export the data to the database.
count = WorksheetFunction.CountBlank(Range("A6:K" & LastRow7))
If count <> 0 Then

    MsgBox "There are some empty cells, you have to introduce data on it/them
        before to export the data"


ElseIf count = 0 Then
```

```
' Check if there is any cell in the wrong format in Sampling Date, Sampling
    Time and Value columns.

    ' Check the Sampling Date format
    For Each cell In Range("G6:G" & LastRow5)
        If IsDate(cell) = False Then
            cell.Interior.ColorIndex = 6 ' Color the cell Yellow
            cell.Value = "BAD FORMAT"
        End If
    Next cell
    ' Check the Sampling Time format
    For Each cell In Range("H6:H" & LastRow6)
        If IsTime(cell) Then
            cell.Interior.ColorIndex = 6 ' Color the cell Yellow
            cell.Value = "BAD FORMAT"
        End If
    Next cell
    ' Check the Value format
    For Each cell In Range("I6:I" & LastRow7)
        If IsNumeric(cell) = False Then
            cell.Interior.ColorIndex = 6 ' Color the cell Yellow
            cell.Value = "BAD FORMAT"
        End If
    Next cell

   iVal = Application.WorksheetFunction.CountIf(Range("G6:I" & LastRow7), "BAD
       FORMAT")

If iVal = 0 Then

MSGL = MsgBox("Are you sure to export the data?", vbYesNo)

If MSGL = vbYes Then

    Dim wb As String
    Dim NewWbk As String
    Dim NewShtName As Variant
    Dim filepath2 As String
    Dim fd As FileDialog

    ' Assign active workbook to variable wb
    wb = ActiveWorkbook.Name

    'Create a FileDialog object as a Save As dialog box.
    Set fd = Application.FileDialog(msoFileDialogSaveAs)

    With fd
        .AllowMultiSelect = False
        .Title = "Save File"
```

```
    .Show
    If .SelectedItems.count > 0 Then
        filepath2 = .SelectedItems(1) & "\"
        NewShtName = .SelectedItems(1)
    Else
        Exit Sub
    End If
End With

Set fd = Nothing

' Add a new workbook
Application.Workbooks.Add
Range("A1").Select

' Save the new workbook and specify location and name
ActiveWorkbook.SaveAs Filename:=NewShtName

' Assign the active workbook to the NewWbk variable
NewWbk = ActiveWorkbook.Name

' Activate the initial window that the macro was run from, and activate the
    Form sheet
Workbooks(wb).Sheets("WaterQuality").Activate

'Select the range to copy
Sheet1.Range("A5:L65536").Select

' Copy the selected range
Selection.Copy

' Activate the new workbbook selecting sheet1
Workbooks(NewWbk).Sheets("Sheet1").Activate

' Paste it into the new workbook
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False

' Rename Sheet1
ActiveSheet.Name = "WaterQuality"

' Save and close New Workbook
ActiveWorkbook.Save
ActiveWorkbook.Close

strCon = "Provider=Microsoft.ACE.OLEDB.12.0; " & _
    "Data Source=C:\Documents and Settings\Administrador\Mis
        documentos\MonEAU\TESTING5 modelEAU Database V.2.accdb"
```

```
    ' Late binding, so no reference is needed
    Set cn = CreateObject("ADODB.Connection")

    cn.Open strCon

    ' Debug.Print ActiveWorkbook.Sheets("sheet1").Name

    ' Insert unto a table called Water_Quality
    scn = "[Excel 8.0;HDR=YES;DATABASE=" & ActiveWorkbook.FullName & "]"
    strSQL = "INSERT INTO WaterQuality " _
           & "SELECT * FROM " & scn & ".[WaterQuality$A5:L" & LastRow7 & "]"

    ' Execute the statement
    cn.Execute strSQL

    cn.Close
    Set rs = Nothing
    Set cn = Nothing

    ' The exportation was successful
    MsgBox "All data is exported successfully on modelEAU Database V.2"

    ' Clear the range
    Sheet1.Range("A6:L65536").ClearContents
Else
    MsgBox "Exportation cancelled"
End If

Else

    MsgBox "There are some cells in Sampling Date, Sampling Time and/or Value
        with a BAD FORMAT."

End If

End If

Else
    MsgBox "There is no the same number of data in Sampling Date, Sampling Time
        and Value"
End If

End If

End Sub

Public Function IsTime(ByVal StrTemp As String) As Boolean

Dim StrShortTime As String
```

```
IsTime = False
StrTemp = Trim(StrTemp)
If StrTemp = vbNullString Then Exit Function

If IsDate(StrTemp) Then
    StrShortTime = FormatDateTime(StrTemp, vbShortTime)
    If StrShortTime = "00:00:00" Then
        If (Not (InStr(1, StrTemp, "0:00:00") > 0)) Or (Not (InStr(1, StrTemp,
            "00:00:00") > 0)) Then
            Exit Function
        Else
            IsTime = True
        End If
    Else
        IsTime = True
    End If
Else
    Exit Function
End If

End Function
```

C.2 Data exportation interface code

R code for the user tools to request the data.

Open the database function

```
####################################################################
############# Open Database and variables definitions #############
####################################################################

# Function to open data base and variable definition

opendb<-function(file,tab){

# Adding columns or add elements to a column function

"%+%"<- function(x,y) paste(x,y,sep="")

# Open the database

library(RODBC)
db <- file.path(file)
channel <- odbcConnectAccess2007(db)
mydata <- sqlFetch(channel, tab)
```

```
# Column lenght

nr<-nrow(mydata)

# Named columns

colnames(mydata)<-c("Project","Responsible","Site","Sample_Point","Description",
"Experiment_Number","Sampling_Date","Sampling_Time","Value","Parameter","Method",
"Comments")

# Time arrangmenet

for (ir in 1:nr)
{
   # split row record
   record_split<-unlist(strsplit(as.character(mydata[ir,"Sampling_Time"]),
   split=" +"))
   mydata[ir,"time"]<-record_split[[2]]
}

mydata[,"Sampling_Time"]<-mydata[,"Sampling_Date"]%+%" "%+%mydata[,"time"]
mydata<-mydata[,-ncol(mydata)]

# Define columns as a factor

fact<-c("Project","Responsible","Site","Sample_Point","Description",
"Experiment_Number","Sampling_Date","Parameter","Method")
nf<-length(fact)
nf_2<-nf*2

for(i in 1:nf){
       mydata[,fact[i]]<-as.factor(mydata[,fact[i]])
}

# Data table

View(mydata)

# Save the table

write.csv2(mydata, file="mydata.csv")

}
```

Doit function

```
# Variables selection function

doit<- function(db,c1,c2,c3,c4,c5,c6,c7){

return (db[c1 & c2 & c3 & c4 & c5 & c6 & c7,])

}
```

Query function

```
####################################################################
######################## Data query function ######################
####################################################################

query <- function(Dat1, Dat2){

# Put together two elements function

"%+%"<- function(x,y) paste(x,y,sep="")

# Charge gWidgets package

options(guiToolkit = "RGtk2")
library(gWidgets)

# Levels definition

Proj <- "='"%+%levels(mydata$Project)%+%"'"
Resp <- "='"%+%levels(mydata$Responsible)%+%"'"
Sit <- "='"%+%levels(mydata$Site)%+%"'"
Point <- "='"%+%levels(mydata$Sample_Point)%+%"'"
Desc <- "='"%+%levels(mydata$Description)%+%"'"
Par <- "='"%+%levels(mydata$Parameter)%+%"'"
Meth <- "='"%+%levels(mydata$Method)%+%"'"

# Define Sampling_Date as a Date/Time field

mydata$Sampling_Date<- as.Date(mydata$Sampling_Date, format="%Y-%m-%d")

# Variables selection function

#doit<- function(db,c1,c2,c3,c4,c5,c6,c7){

#return (db[c1 & c2 & c3 & c4 & c5 & c6 & c7,])

#}
```

```
# Make object within dataframe accessible

attach(mydata)

# Pick lists GUI

lst <- list()
lst$action <- list(beginning="doit(mydata,",ending=")")
lst$assignto<-T
lst$arguments$Query$Project<- list(type = "gcombobox",items=Proj)
lst$arguments$Query$Responsible<- list(type="gcombobox", items=Resp)
lst$arguments$Query$Site<- list(type = "gcombobox",items=Sit)
lst$arguments$Query$Sample_Point<- list(type = "gcombobox",items=Point)
lst$arguments$Query$Description<- list(type="gcombobox", items=Desc)
lst$arguments$Query$Parameter<- list(type = "gcombobox",items=Par)
lst$arguments$Query$Method<- list(type="gcombobox", items=Meth)

ggenericwidget(lst, container=gwindow("Scatterplot"))

# Select date range

Dat1 <- as.Date(Dat1)
Dat2 <- as.Date(Dat2)

var1 <- subset(var1, var1$Sampling_Date>=Dat1 & var1$Sampling_Date<=Dat2)
var2 <- subset(var2, var2$Sampling_Date>=Dat1 & var2$Sampling_Date<=Dat2)
var3 <- subset(var3, var3$Sampling_Date>=Dat1 & var3$Sampling_Date<=Dat2)
var4 <- subset(var4, var4$Sampling_Date>=Dat1 & var4$Sampling_Date<=Dat2)

# Clean up the dataset

#detach(mydata)

}
```

Plot function

```
####################################################################
############################### Plot data #########################
####################################################################

detach(mydata)
plotgraph <- function(n, m){

# Adding columns or add elements to a column function

"%+%"<- function(x,y) paste(x,y,sep="")
```

```
if (n == 1) {

# Define dataframe for the selected values

leg1<-var1[1,"Parameter"]%+%"-"%+%var1[1,"Method"]
graph1<-var1[,c("Sampling_Time","Value")]

View(graph1)

# Define Sampling_Time as a Date/Time field

graph1$Sampling_Time<- as.POSIXct(graph1$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")

# Plot selected values

plot(graph1$Sampling_Time, graph1$Value, type= "p" , xlim= NULL, col = "red",
size =1, xlab= "Time",ylab= leg1, main= " ",format="%Y-%b-%d %H:%M")
legend("topright", legend = (leg1), col=("red"),lty=c(1,1),lwd=5, cex=.5)

# Save plot as a png file

dev.copy(png,'graph.png')
dev.off()

}

if (n == 2) {

# Define dataframe for the selected values

leg1<-var1[1,"Parameter"]%+%"-"%+%var1[1,"Method"]
leg2<-var2[1,"Parameter"]%+%"-"%+%var2[1,"Method"]
graph1<-var1[,c("Sampling_Time","Value")]
graph2<-var2[,c("Sampling_Time","Value")]

#View(graph1)
#View(graph2)

# Define Sampling_Time as a Date/Time field

graph1$Sampling_Time<- as.POSIXct(graph1$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph2$Sampling_Time<- as.POSIXct(graph2$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")

if (m ==1){
```

```
# Plot selected values at the same axis

plot(graph1$Sampling_Time, graph1$Value, type= "p" , col = "red", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time)),
ylim=range(c(graph1$Value,graph2$Value)))
par(new=TRUE)
plot(graph2$Sampling_Time, graph2$Value, type= "p" , col = "blue", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time)),
ylim=range(c(graph1$Value,graph2$Value)))
legend("topright", legend = c(leg1,leg2),
    col=c("red","blue"),lty=c(1,1),lwd=5, cex=.5)

# Save plot as a png file

dev.copy(png,'graph.png')
dev.off()
}

if (m == 2) {

# Plot selected values in two different axis

par(mar=c(5,4,4,5)+.1)
plot(graph1$Sampling_Time, graph1$Value,type= "p" ,col = "red", size =1,
xlab= "Time", ylab= leg1, main= " ",format="%Y-%b-%d %H:%M")
axis(2, col="black")
par(new=TRUE)
plot(graph2$Sampling_Time, graph2$Value, axes=F, type= "p",col = "blue", size
    =1,
xlab= "Time", ylab= leg1, format="%Y-%b-%d %H:%M")
axis(4, col="black")
mtext(leg2,side=4, line=3)
#axis(1,pretty(range(time),10))
#mtext(side=1,col="black",line=)
legend("topright", legend = c(leg1,leg2),
    col=c("red","blue"),lty=c(1,1),lwd=5, cex=.5)

# Save plot as a png file

dev.copy(png,'graph.png')
dev.off()

}

}

if (n == 3) {
```

```
# Define dataframe for the selected values

leg1<-var1[1,"Parameter"]%+%"-"%+%var1[1,"Method"]
leg2<-var2[1,"Parameter"]%+%"-"%+%var2[1,"Method"]
leg3<-var3[1,"Parameter"]%+%"-"%+%var3[1,"Method"]
graph1<-var1[,c("Sampling_Time","Value")]
graph2<-var2[,c("Sampling_Time","Value")]
graph3<-var3[,c("Sampling_Time","Value")]

# Define Sampling_Time as a Date/Time field

graph1$Sampling_Time<- as.POSIXct(graph1$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph2$Sampling_Time<- as.POSIXct(graph2$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph3$Sampling_Time<- as.POSIXct(graph3$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")

# Plot selected values

plot(graph1$Sampling_Time, graph1$Value, type= "p" , col = "red", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time, graph3$Sampling_Time)),
ylim=range(c(graph1$Value,graph2$Value,graph3$Value)))
par(new=TRUE)
plot(graph2$Sampling_Time, graph2$Value, type= "p" , col = "blue", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time, graph3$Sampling_Time)),
ylim=range(c(graph1$Value,graph2$Value,graph3$Value)))
par(new=TRUE)
plot(graph3$Sampling_Time, graph3$Value, type= "p" , col = "green", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time, graph3$Sampling_Time)),
ylim=range(c(graph1$Value,graph2$Value,graph3$Value)))
legend("topright", legend = c(leg1,leg2,leg3), col=c("red","blue","green"),
lty=c(1,1),lwd=5, cex=.5)

# Save plot as a png file

dev.copy(png,'graph.png')
dev.off()

}

if (n == 4) {

# Define dataframe for the selected values
```

```
leg1<-var1[1,"Parameter"]%+%"-"%+%var1[1,"Method"]
leg2<-var2[1,"Parameter"]%+%"-"%+%var2[1,"Method"]
leg3<-var3[1,"Parameter"]%+%"-"%+%var3[1,"Method"]
leg4<-var4[1,"Parameter"]%+%"-"%+%var4[1,"Method"]
graph1<-var1[,c("Sampling_Time","Value")]
graph2<-var2[,c("Sampling_Time","Value")]
graph3<-var3[,c("Sampling_Time","Value")]
graph4<-var4[,c("Sampling_Time","Value")]

# Define Sampling_Time as a Date/Time field

graph1$Sampling_Time<- as.POSIXct(graph1$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph2$Sampling_Time<- as.POSIXct(graph2$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph3$Sampling_Time<- as.POSIXct(graph3$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")
graph4$Sampling_Time<- as.POSIXct(graph4$Sampling_Time, format="%Y-%m-%d
    %H:%M:%S")

# Plot selected values

plot(graph1$Sampling_Time, graph1$Value, type= "p" ,col = "red", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time,graph3$Sampling_Time,
graph4$Sampling_Time)),ylim=range(c(graph1$Value,graph2$Value,
graph3$Value,graph4$Value)))
par(new=TRUE)
plot(graph2$Sampling_Time, graph2$Value, type= "p" , col = "blue", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time,graph3$Sampling_Time,
graph4$Sampling_Time)),ylim=range(c(graph1$Value,graph2$Value,
graph3$Value,graph4$Value)))
par(new=TRUE)
plot(graph3$Sampling_Time, graph3$Value, type= "p" , col = "green3", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time,graph3$Sampling_Time,
graph4$Sampling_Time)),ylim=range(c(graph1$Value,graph2$Value,
graph3$Value,graph4$Value)))
par(new=TRUE)
plot(graph4$Sampling_Time, graph4$Value, type= "p" , col = "yellow", size =1,
xlab= "Time",ylab= "Value", main= " ",format="%Y-%b-%d %H:%M",
xlim=range(c(graph1$Sampling_Time,graph2$Sampling_Time,graph3$Sampling_Time,
graph4$Sampling_Time)),ylim=range(c(graph1$Value,graph2$Value,
graph3$Value,graph4$Value)))
legend("topright", legend = c(leg1,leg2,leg3,leg4),
col=c("red","blue","green","yellow"),lty=c(1,1),lwd=5, cex=.5)

# Save plot as a png file
```

```
dev.copy(png,'graph.png')
dev.off()

}

}
```