Proceedings of the 11th International Coral Reef Symposium, Ft. Lauderdale, Florida, 7-11 July 2008
Session number 16

COREweb, a web-based information management solution for experimental data from the field of coral reef ecology

F.W. Mayer[1], S. Duewel[2], A. Haas[1], M. Naumann[1], C. Jantzen[1], J.M. Jeschke[3,4], C. Wild[1]

1) Coral Reef Ecology Work Group (CORE), GeoBio-Center & Department of Earth and Environmental Science, Ludwig-Maximilians-Universität (LMU) München, Richard-Wagner-Str.10, 80333 München, Germany
2) proCon IT GmbH, Wamslerstr. 9, 81829 München, Germany
3) Dept. Biology II, Evolutionary Ecology, LMU München, Grosshaderner Str. 2, 81852 Martinsried-Planegg, Germany
4) Cary Institute of Ecosystem Studies, P.O. Box AB, Millbrook, NY 12545, USA

Abstract. In the profit-oriented business and industry sector, information management (IM) solutions are well established. But in coral reef ecology, the IM aspects of experimental design, data acquisition and evaluation, documentation and publication follow the individual workers' experience rather than established standards. Although these tasks are largely automatable, no useable guidance tool has been established so far that would lead the worker through the experimental life cycle and enforce validation, documentation, completeness, correctness and consistency of data. Often, this causes loss of information and diminishes data quality as well as compatibility. We developed COREweb, a dynamic web application to automate the scientific work process in consistency with established scientific work practices. The underlying data model is capable of managing any kind of manipulative or monitoring ecological experiment. The user operates COREweb through an intuitive Graphical User Interface and focuses on the scientific experimentation and validation process. Data can be shared among team members. COREweb is scalable from a single-user, single-desktop environment up to a distributed network, connecting teams worldwide. As COREweb is easily adapted to many different scientific experiments, it can serve as a prototype model for useable scientific IM tools that make valuable data better available to decision makers.

Keywords: Scientific work flow, process model, data model, information management

Introduction

In life sciences, researchers proximately want to publish as fast and high-ranked as possible with the least effort of time, financial and personal resources. Therefore, any means of facilitating and improving the scientific work flow should be in their best interest. Information management (IM) is the collection and management, *i.e.* the organisation and control of the structure, processing and delivery of information. Information in research projects consists of the experimental data itself, the metadata, instructions on methodology, best practices, abided protocols and standards as well as any uncategorized information which ultimately helps a research team to gain knowledge. A rising number of collaborators, close and remote, requires means of online collaboration and tools to share data and knowledge. The following tools and standards exist for this purpose.

Metadata and data standards The ecological markup language (EML) is an extensible markup language (XML) extension that serves to document and handle ecological metadata (http://knb.ecoinformatics.org/software/eml/). Metadata are needed for the identification and documentation of the multitude of heterogeneous ecological and environmental data sets. However, proper documentation via metadata depends on the individual worker's motivation and is therefore not very prevalent.

In some software packages, such as ECOBAS (Benz et al. 2001), data completeness, correctness and consistency is automatically enforced by the package's functionality.

Experimental design and data modelling Calvin Dytham presented a simple-to-use decision tree to guide the scientist from the formulation of their research questions towards the choice of an appropriate statistical test (Dytham 2003).

The software *Touchstone* allows the user to refine the experimental design in a "what if" style until a statistically satisfying setup has been found (Mackay et al. 2007).

Pratt (1995) designed a relational database schema which can store the data of empirical studies. The schema bases on a model of an empirical study of natural processes.

Work flows and method provenance Kepler automates work flows, i.e. the processing steps from raw to analyzed data (Bowers et al. 2006), and features a model for user-oriented data provenance in pipelined scientific workflows. Provenance information may also be used by scientists to reproduce results from earlier runs, to explain unexpected results, and to prepare results for publication. Bowers et al. (2006) developed a simple provenance model that is capable of supporting a wide range of applications even for complex models of computation, such as process networks.

The *Karma* provenance framework provides a means to collect workflow, process, and data provenance from data-driven scientific workflows (Simmhan et al. 2006) in order to retain the knowledge that was put into the experimental planning and refinement.

Data lineage Bose and Frew (2005) review the data quality benefits of lineage, i.e. the documentation of data origin and processing history. Lineage helps to understand data processing steps (Woodruff et al. 1997), enhances interpretation, prevents misinterpretation of data and communicates data suitability, reliability, accuracy, currency and redundancy (Eagan et al. 1993). Lineage also facilitates the use of historical data (Clarke and Clark 1995).

Process models In software design, several procedures exist which structure and guide the process of coding new programs. The V-model (Boehm 1979) is one approach that parallels ecological experiments.

In the V-model, the initial steps of conceptual planning preceding the actual writing of source code are designed with regard to the requirements of the later steps of testing and evaluating the software. The writing of source code begins only once its desired functionality is defined and understood.

Since the V-model's "think-first" approach helps eliminating ambiguity and identifying non-trivial requirements, the time spent with planning ahead pays off at the stage of generating the source code.

Similarly, the researcher has to design manipulation and observation routines with regard to the statistical power of the generated data in order to avoid generating data with a low statistical power and, therefore, losing valuable time and financial resources.

Aim of this study This study aims to incorporate these established standards and best practices into a process model of an ecological experiment by abstracting the ecological work process and its related artefacts into formal languages. This is achieved through the graphical user interface (GUI) structure and database schema of a database-driven web application, which provides modular tools that guide and assist the user through the experiment's life cycle.

Materials and Methods

Via model-driven development (MDD), COREweb was generated to a large extent from the formal language artefacts using Grails, a MDD framework (Rocher et al. 2006). COREweb is a model-view-controller (MVC) web application, which provides a separation of the database structure (model), the GUI (view) and the functionality (controller).

The COREweb prototype currently supports the process up to the generation of the data structure. It is possible to setup and thoroughly document the experiment, enter the raw data with input validation at a user-defined level and export the validated, well documented data and metadata for processing and statistical analysis.

COREweb is deployed as a Java applet, connected to a postgres database, runs on a web server and can be accessed in any browser via the inter- or intranet. The database and the java applet can run on one or on different machines. Source code, technical documentation and a walkthrough using an example experiment are available at http://palmuc.de/core/coreweb/ .

Results

Abstracting the experiment into a database schema

Methods consist of several manipulative or observational steps. Samples are the individual experimental units, which are manipulated and observed. Observation results in the measurement of their properties.

A sample experiment is shown in Fig. 1; two methodical steps result in a total of five different measurements. The sum of measurements and the number of samples define the data structure, paralleling the columns (measurements) and rows (samples) of a spreadsheet.

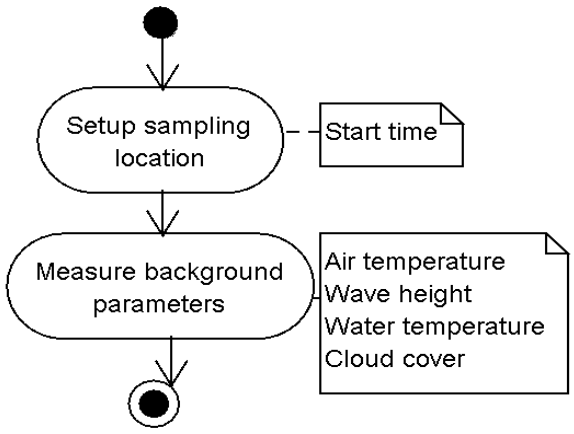

Figure 1: A sample experiment with two methodical steps (round boxes) and five measured parameters (rectangular boxes).

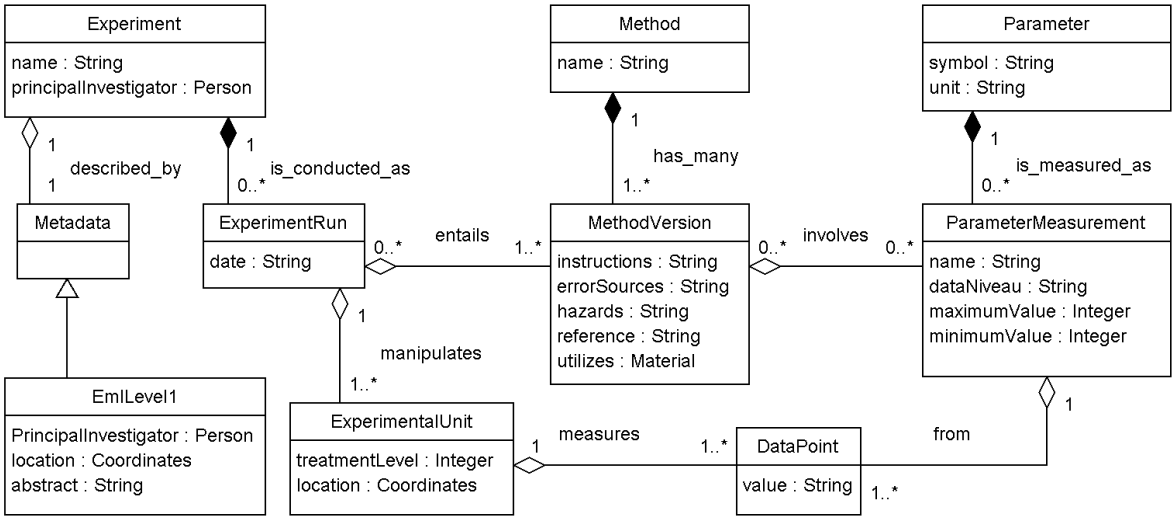

Figure 2: Simplified data model of an abstracted scientific experiment, shown as a UML class diagram. Each class, represented by a box, corresponds to a data table in the database backend. Each relation between classes, represented by a connecting line, corresponds to a foreign key in the database backend.

A database schema was designed that can store information on metadata, documentation, methodology, information on experiments, their individual realizations (runs), the samples and the measured parameters by only adding entries to the existing tables (Fig. 2).

In contrast to conventional spreadsheets, the data points are linked to their methodical origins (MethodVersion) and metadata documentation, providing data lineage, method provenance and semantics via metadata.

The database backend itself is hidden from the user, who interacts with the application through an intuitive GUI front end (Fig. 3), which unobtrusively offers best-practice functionalities at the appropriate process stages of the experiment and implements standards, e.g. the EML metadata format. MDD allowed a large portion of code to be generated from the database schema.

Process model of an ecological experiment Paralleling the V-model, a proposed work flow of an ecological experiment, based on manipulation and observation, was modelled in the unified modelling language (UML, information and tutorials at **http://www.uml.org/** , Fig. 4). It features iterative testing using automatically generated dummy data, automation of the data processing and evaluation steps and refinement of the experimental design. It focuses on the statistical power of the generated data.

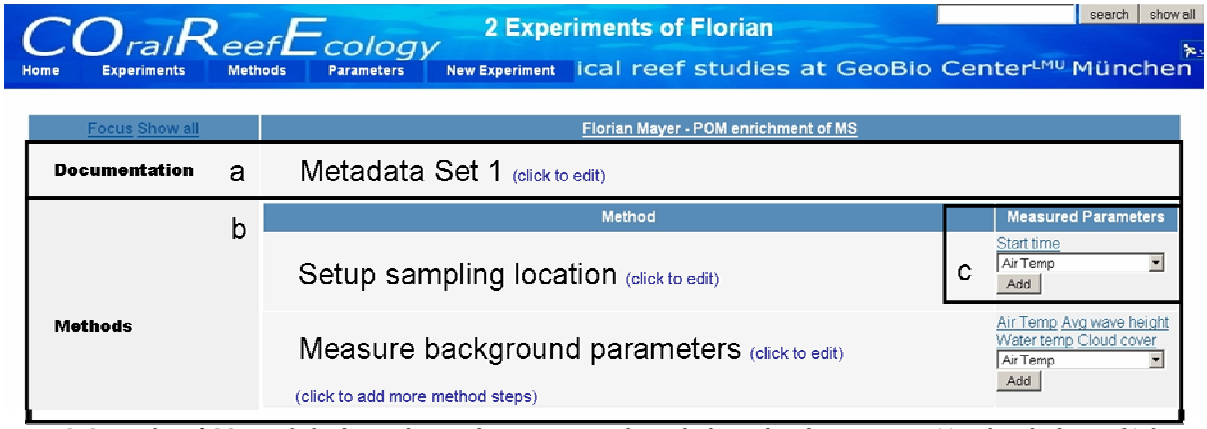

Figure 3: Screenshot of COREweb displaying the sample experiment with attached metadata documentation (a) and method steps (b) during which several parameters (c) are measured.

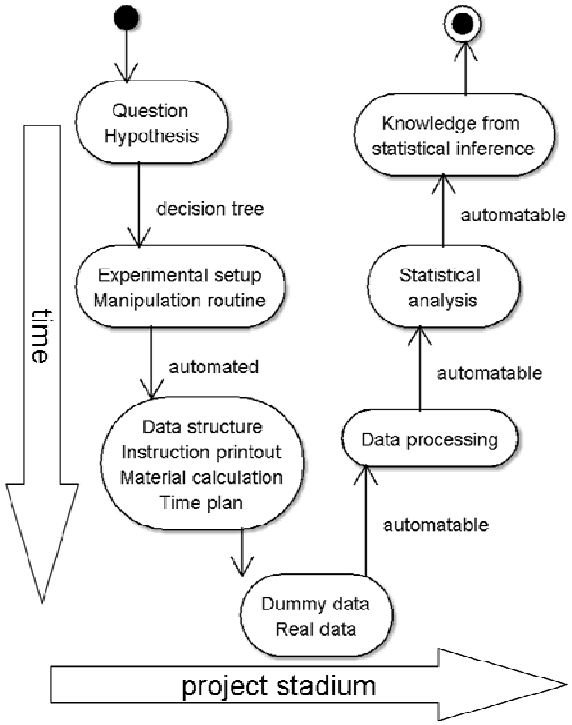

Figure 4: Process model of an experiment. The process can be dry-run and refined using dummy data. Real data should be generated after the "dry-run" validation of the process.

The proposed work flow consists of:

(1) The initial description of the planned experiment and formulation of the questions and hypotheses. Input masks channel this information into valid EML metadata; reporting functions produce an initial project outline from the entered information.

(2) Choosing the appropriate statistics to test the hypotheses following a decision tree adapted from Dytham (2003), leading to a possible refinement of the hypothesis and, ultimately, to the statistical constraints on the experimental setup.

(3) Choosing and documenting the experimental setup, the manipulation and sampling methods. Export functions provide methodical instructions, material requirement lists, a Gantt-chart style time plan, the sampling scheme, lineage documentation and the resulting data structure in form of a data table. Figure 5 shows the data table resulting from the sample experiment shown in Fig. 1. At this stage, the user can map the statistical constraints (e.g., one grouping variable, one affector and one responder variable *sensu* Pratt (1995)) onto the respective parameters.

(4) Validation of the planned experiment using automatically generated dummy data in form of a separate "experimental run." Implementation of an export into comma separated value (CSV) spreadsheets is pending.

(5) Transformation of raw dummy or real data into processed, analyzable data. This step is potentially automatable via workflows (e.g. using *Kepler*). Bulk import of existing data, such as sensor data, will be implemented in this step.

(6) Statistical analysis of the data for inference and evaluation of the hypotheses. Pipelining the data and desired statistics as input arguments; the analysis in the software package R can be automated as well as the export to CSV spreadsheets. As the least common denominator file format, CSV can be read by most analysis packages.

(7) Interpretation of the statistical output and transformation of the analysis into knowledge.

Up to this stage, dry-testing the process with dummy data entails continued documentation and refinement of the experiment. As soon as the experiment is validated and the statistical power of the experimental setup is satisfactory, the process steps 1 - 7 can be used to conduct the real experiment. The EML metadata will be made exportable to a readable table format using style sheets as well as to established metadata databases.

Service-oriented architecture (SOA) The single work steps are implemented as optional services, so the user can choose between guidance and self-determination. COREweb functions as a framework with plug-ins of existing technology; the least common denominator data structure (see Fig. 2) makes data and information compatible between the modules.

	Setup sampling location - Start time	Measure bg parameters - Air temp	Measure bg parameters - Wave ht	Measure bg parameters - Water temp	Measure bg parameters - Cloud cover
Sample 1					
Sample 2					
Sample 3					
Control 1					

Figure 5: The resulting data table from the sample experiment (see Figs. 1 and 3). The column headers are tagged with the respective method and parameter names.

Discussion

Advantages of software-conveyed guidance The database schema of COREweb is able to store raw data from ecological experiments. In combination with appropriate metadata documentation, results compatibility and transparency of data sets.

Data completeness, correctness and consistency can be easily enhanced by the GUI's input validation. For example, by simply filling in the metadata form fields, the user creates valid EML metadata without having to know EML. Similarly, the user can determine the data type of the measured parameters. From these constraints, the software can validate input, e.g. warn if characters are entered in a date field.

Automation and simulation via dummy data encourages the user to simulate the experiment in advance and to subsequently refine the planned process. This may save time and financial resources, if e.g. the number of samples can be optimized.

Early documentation creates a quick overview of the planned methodology. Incremental refinement of both the documentation and the methodology provenance *sensu* Bose and Frew (2005) provide advantages for the team in terms of consistent methodology and knowledge exchange. It also leads to raised data compatibility and quality.

Assigning the statistical relevant terms (e.g., grouping variable, affector, responder) onto the data structure helps to document the experiment's semantics.

Following a user interface's work flow implicitly and unobtrusively guides the user along the implemented standards and best practices.

Challenges for a computerized IM solution Mueller (1994) after early studies with computerized heuristic tools for engineers found that "it is not in the human nature to work rationally dominated". Until now, no IM solution successfully integrates standards and best practices into the scientific work process. This may indicate that on the one hand, known benefits of adhering to those standards and best practices do not feed back quickly or tangible enough into a speed-up or improvement of the user's own work process. On the other hand, Jagadish et al. (2007) emphasized the lack of usability of recent IM software. For non-scientific use however, intuitive and very well useable web applications exist, such as social networking platforms, which collect user-generated data and provide connectivity by addressing the users' play instinct and acquisitiveness. COREweb tries to overcome the user's initial reluctance with tangible rewards, e.g. documentation spin-offs like material calculation lists, time plans and field instructions that can be set up quickly and refined subsequently as well as with a good usability.

Mueller (1994) also found that his test subjects did not use software solutions voluntarily. Our SOA approach leaves most guiding functionalities optional; therefore, the user retains maximum autonomy.

Although the GUI of COREweb was purposefully kept simple in order to focus on development of the key functionalities, the ongoing implementation as an MVC web applet allows a refinement of the GUI towards a smooth and increasingly intuitive interface.

Summary

The software prototype COREweb implements core functionalities of the scientific work flow, implicitly integrating established standards and best practices.

Via automation of the data processing and analysis steps, COREweb facilitates a test-first approach using dummy data. Documentation rewards the user through the feedback loop of useful printout artefacts, such as material calculation lists, time plans or work instruction sheets.

Further development of COREweb will increase usability and smoothness of the work flow as well as it will implement more functionality.

Acknowledgements

This work was supported by a PhD stipend to Florian Mayer from Fazit Foundation, Frankfurt, Germany. The authors want to thank Axel Rauschmayer (LMU Munich), Trina Myers and Dr. Ian Atkinson (JCU, Townsville) for fruitful discussions and valuable input.

References

Benz J, Hoch R, Legovic T (2001) Ecobas - modelling and documentation. Ecol Model 138:3–15

Boehm B (1979) Guidelines for verifying and validating software requirements and design specifications. In EURO IFIP 79:711–719

Bose R, Frew J (2005) Lineage retrieval for scientific data processing: A survey. Acm Comp Surv 37:1–28

Bowers S, McPhillips T, Ludaescher B, Cohen S, Davidson S (2006) A model for user-oriented data provenance in pipelined scientific workflows. In Provenance and annotation of data. Springer, Berlin, pp 133–147

Clarke D, Clark D (1995) Lineage. In Guptill SC, Morrison JL (ed) Elements of spatial data quality. Elsevier, Oxford, pp 13–30

Dytham C (2003) Choosing and using statistics: A biologist's guide. Blackwell Publishing, Oxford

Eagan P, Ventura S (1993) Enhancing value of environmental data: Data lineage reporting. J Env Eng 119:5.

Jagadish H, Chapman A, Elkiss A, Jayapandian M, Li Y, Nandi A , Yu C (2007) Making database systems usable. In Proc 2007 ACM SIGMOD ICMD. ACM Press, New York, pp 13–24

Mackay WE, Appert C, Beaudouin-Lafon M, Chapuis O, Du Y, Fekete JD, Guiard Y (2007) Touchstone: exploratory design of experiments. In Proc 2007 SIGCHI Conf Hum Fact Comp Sys. ACM Press, New York, pp 1425–1434

Mueller J (1994) Akzeptanzprobleme in der Industrie, ueber Ursachen und Wege zu ihrer Ueberwindung. In Pahl G (ed) Psychologische und paedagogische Fragen beim methodischen Konstruieren. TUEV Rheinland, pp 247–266

Pratt, J. (1995). Data modeling of scientific experimentation. In Proc 1995 ACM Symp Appl Comp, pp 86–90

Rocher G, Laforge G, Koenig D (2006) The definitive guide to grails. Apress, Berkeley

Simmhan Y, Plale B, Gannon D (2006) A framework for collecting provenance in data-centric scientific workflows. In Proc IEEE ICWS'06, pp 427-436

Woodruff A, Stonebraker M (1997) Supporting fine-grained data lineage in a database visualization environment. In Proc 13th ICDE, pp 91–102