

Modeling the fate of down-the-drain chemicals in rivers: An improved software for GREAT-ER

Koormann, F. ^{a,*} Rominger, J. ^{b,**} Schowanek, D. ^c Wagner, J.-O. ^a
Schröder, R. ^d Wind, T. ^e Silvani, M. ^f Whelan, M.J. ^g

^a*Intevation GmbH, Georgstrasse 4, 49074 Osnabrück, Germany*

^b*TechniData AG, Environmentals, Dornierstr. 3, 88677 Markdorf, Germany*

^c*Procter & Gamble BIC, Temselaan 100, 1853 Strombeek-Bever, Belgium*

^d*Henkel, WEQ/QM + Environment, Düsseldorf, Germany*

^e*Henkel, VTB-Ecology, Düsseldorf, Germany*

^f*BASF AG, GUP/CA - Z470, Ludwigshafen, Germany*

^g*Unilever Colworth Laboratory, Safety and Environmental Assurance Centre,
Sharnbrook, United Kingdom*

Abstract

GREAT-ER (Geography-referenced Regional Exposure Assessment Tool for European Rivers) is a model system for predicting chemical fate and exposure in surface waters. The GREAT-ER approach combines a series of well studied models (for sewers, waste water treatment plants and rivers) with spatial information managed by a GIS. A new version of GREAT-ER (version 2) has been developed which has a number of improved features compared to its predecessor. All components are now implemented as Free Software and make use of Free Software where possible. This simplifies the distribution, use and extension of the system and its components for scientific applications. In addition, two new platform-independent user interfaces have been developed. The first is a simplified web-based interface for easy access without installation and the second is a full-featured desktop version. The model system has been redesigned to increase its flexibility for the modification and extension of its component models. In addition, the file-based data storage system used in version 1 has been replaced by a database management system with a flexible Application Programming Interface (API). This makes it easier for users to share their results, using a new repository (also with web-based and desktop versions) serving as a central communication tool. The output from the new system has been successfully tested against the field-validated models in GREAT-ER 1.0.

Key words: Environmental Modeling, Exposure Assessment, GIS, Database, GREAT-ER

1 Introduction

Environmental risk assessments for chemicals generally make comparisons between predicted environmental concentrations (PECs) and predicted no effect concentrations (PNECs). Standard assessments are generally performed for “generic” scenarios which quantify relative risk (e.g. European Commission, 2003). However, actual risk will vary temporally and spatially with factors such as river and wastewater flow and loss processes during transport. Thus more realistic predictions of exposure require more sophisticated models which account for these factors. “Down-the-drain” chemicals are those substances used as ingredients in domestic consumer products (e.g. detergent ingredients, pharmaceuticals) which are usually disposed of via wastewater systems after use. A crude PEC in sewage can be calculated by dividing the per-capita mass of substance used by the domestic per-capita water use. This can then be adjusted for any removal during transport in the sewer network and in sewage treatment and for dilution in the receiving water body to give PEC for surface water, immediately after mixing but before any degradation has taken place. Further adjustments can be made by simulating the processes operating in the stream channel. However, river basins are complex systems with a wide variety of land use, population density and industrial activity containing stream and river channels which vary greatly in terms of geometry, flow rate and water quality. This complexity and variability can best be represented in a Geographic Information System (GIS) and the spatially referenced data therein used to predict spatially and temporally varying concentrations using simulation models, describing transport and transformation processes (e.g. Koormann et al., 1998; Shena, 2005).

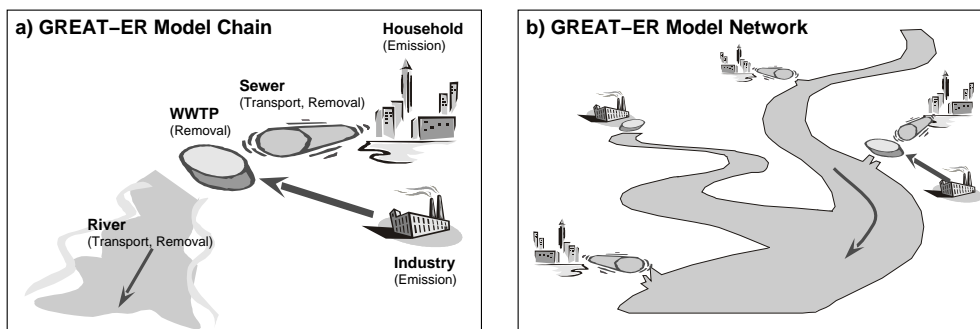


Fig. 1. Schematic view of a) GREAT-ER model chain and b) GREAT-ER model network as combination of multiple model chains

The European project GREAT-ER (Geography-referenced Regional Exposure Assessment Tool for European Rivers, 1996 - 1999 (Feijtel et al., 1997; Schowanek et al., 2001)) was launched as an international effort to develop and validate such a model system: Models for the sewer, waste water treatment plant (WWTP) and

* frank.koormann@intevation.de

** jochen.rominger@technidata.de

river compartments (Boeije et al., 1997; Trapp and Matthies, 1998; Boeije, 1999) were coupled into model chains (Figure 1 a). To consider spatial aspects like channel geometry or multiple discharges several local model chains were linked into a model network (Figure 1 b). Temporal variability in state variables such as chemical loads, river discharge and river velocity was accounted for by performing a large number of steady-state deterministic model iterations using Monte-Carlo simulation (in which values of key variables are randomly selected from pre-defined probability density functions). The statistics (e.g. the mean or a given percentile) of the spatially-referenced PECs generated by the model for a particular river network can be viewed as color coded maps using the GIS interface (Boeije et al., 2000).

As part of the GREAT-ER project monitoring studies were conducted in five catchments in the UK and Italy and the results from these studies were used to validate the PECs produced by the GREAT-ER model (Holt et al., 2000; Gandolfi et al., 2000). The system has subsequently been applied to many other catchments in Germany, Belgium, France and other European countries (Verdonck et al., 2000; Schulze and Matthies, 2001; Wagner, 2001). These studies have consistently proven the applicability of the GREAT-ER concept in the prediction of “down-the-drain” chemical fate and exposure in aquatic systems.

A first version of GREAT-ER (version 1.0) was implemented within the project. This prototype version has several limitations: One of the disadvantages with the GREAT-ER 1.0 software was that it was implemented using a proprietary GIS package (ESRI ArcView) to visualise spatially-referenced inputs and outputs. Although GREAT-ER itself was distributed free of charge, users without existing access to the GIS package were forced to incur a significant cost in order to use it. In addition, a file-based data storage system was used which limited data security and confidentiality. Furthermore, the model system was not particularly transparent which made future enhancements (e.g. the addition of new models or compartments) difficult.

This paper describes a major re-design of the GREAT-ER software (GREAT-ER 2.0) which was developed in a modular fashion around a database management system (storing spatially-referenced and substance-specific data). GREAT-ER 2.0 incorporates a detailed access concept, a scaled set of user interfaces with GIS capabilities and a model server providing data operations via the interfaces. It also contains a flexible tool for preparing and quality controlling the requisite input data. It should be noted that no changes have been made to the model equations used by GREAT-ER which are adequately described elsewhere (e.g. Boeije et al., 1997; Trapp and Matthies, 1998; Boeije, 1999).

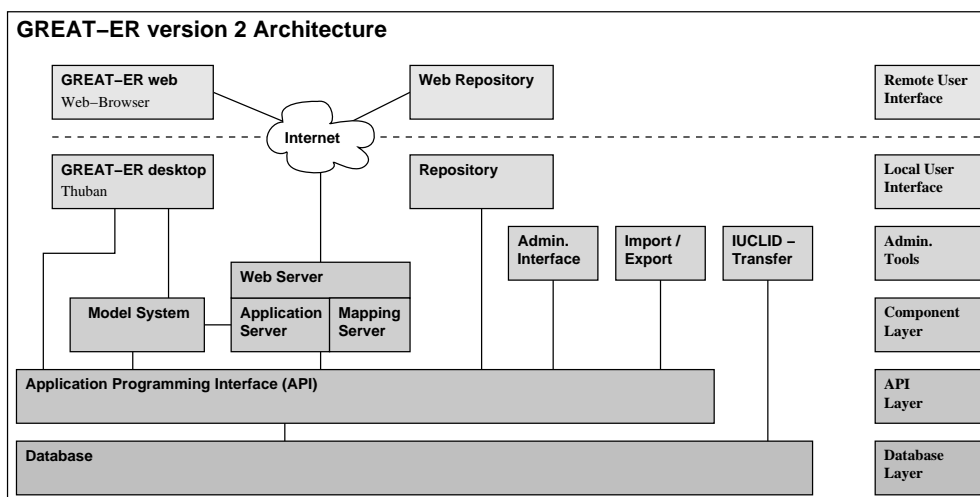


Fig. 2. Software architecture of GREAT-ER 2

2 GREAT-ER version 2 Architecture

Figure 2 illustrates the architecture of the new GREAT-ER. The entire system is founded on a database management system, the Application Programming Interface (API) provides an additional abstraction layer. A specific administration tool offers functionality to manage the relevant parts of the GREAT-ER database including import / export of simulation data. An interface to the International Uniform Chemical Information Database (IUCLID, Boos (2001)) provides an easy option to upload substance specific data (such as physico-chemical properties and degradation rate constants) into GREAT-ER from a central source. Two variants of a repository (see 5.3) provide access to the parameters from a database view and the management of related documents. The model server performs simulation runs issued by the two user interfaces. All data exchange between model server and user interfaces is handled via the database.

Two user interfaces have been implemented in GREAT-ER 2. A desktop interface (GREAT-ER desktop) has been constructed based on a Free Software geographic data viewer (Thuban¹) which connects directly to database and model server. In addition, a web browser-based user interface (GREAT-ER web) has been built and is provided via a web server with an underlying application server (Zope²) for application and session management and a mapping server (UMN MapServer³) providing map rendering and navigation.

In comparison to the desktop version, GREAT-ER web offers simplified functionality but on the other hand does not require further software installation or licenses

¹ <http://thuban.intevation.org>

² <http://www.zope.org>

³ <http://mapserver.gis.umn.edu>

on the user side. As commonly used in modern modeling systems (e.g. Halls, 2003; Bellasio, 2005) GREAT-ER web serves as an entry point to the GREAT-ER concepts and system.

GREAT-ER desktop offers the full functionality of the GREAT-ER system with various analysis tools and import/export features. It can be used on a single computer (local database and modeling system), but can also benefit from a network with databases shared within work groups and available computing servers.

3 Database

3.1 API abstraction layer

The GREAT-ER API couples the different GREAT-ER user interfaces with the database in which all data are stored. This middle-ware component consists of a dynamic link library written in C/Embedded SQL offering all required database operations in a standardized form. A wide set of functionality to read, create, actualize, delete and search data is implemented and encapsulates the internal complexity of database operations. The API also restricts client application actions on the database to a set of valid operations to increase data safety and realizes the user privilege concept.

3.2 Privileges

GREAT-ER is designed as multi-user system. Users may work on a central system with different data and decide which parts of the data share "public" characteristics and which are confidential or write protected. In this way use restrictions and data ownership can be controlled.

Each GREAT-ER object (catchment, session, substance, environmental data, binary object) is supplemented with the following information:

- Identification of object owner
- State of the data (meta information. e.g. valid/invalid)
- Creation and last modification date
- Privileges (all, group, owner).

This information serves as basis for the GREAT-ER authorization concept. For each GREAT-ER object the creator automatically becomes the data owner and can select between several privilege levels.

Users are split into three types:

- (1) object owner
- (2) members of the object owner's user group
- (3) all other users

The object owner can provide the object with view or write privileges for the three user types owner, user group and all. Authorization groups and users instead can only be defined by an administrator inside of the administration utility. This concept allows flexible but also restricted management of GREAT-ER data for many users.

3.3 Flexible Parameter Data Structures

GREAT-ER 2 provides a completely new and flexible approach for the definition of all kinds of model and system parameters. A parameter is usually described by name, unit, warning and error range, type, comment and more ("data dictionary"). The use of a "parameter tree" concept enhances this description by introducing a hierarchy and sorting information on levels of blocks, groups and fields. It is used to define, dynamically, parameter trees, which show e.g. substance parameters logically grouped into biodegradation, partitioning, instream removal and others.

The most obvious advantage of this kind of parameter management is its potential for extending the system. It is possible to add new parameters, or change existing ones without having to change the underlying database architecture.

4 Model System

GREAT-ER implements a model system consisting of four sub-models: emission, sewer, WWTP and river. There are up to three complexity modes for each of the sub-models. These range from the application of simple elimination rates derived from field studies to process models based on laboratory and basic physico-chemical data.

All of the sub-models are deterministic in principle. Using the Monte-Carlo method on top of these models, GREAT-ER implements a stochastic approach and results are returned as distributions of concentrations in each river stretch and for each discharge site (Feijtel et al., 1997; Boeije et al., 1997). This approach is consistent with the method of combining distributions advocated by Warn and Brew (1980) and Warn (1982) and will result in superior estimations of the statistical moments of river water concentrations compared with deterministic applications of the mass balance equation. It is also implemented in operational water quality models like SIMCAT, TOMCAT and RQP (SEPA, 2002).

In addition to performing simulations, the model system provides commands for obtaining extra information about the underlying models:

- **Required Parameters:** List all parameters required by the model to assist the user to identify the parameters to be filled in.
- **Missing Parameters:** List all parameters still missing for the model to assist the user in filling in a complete parameter set.
- **Stochastic Parameters:** Those parameters which can be described by a probability density function in the model. Based on the current settings the model derives the list of parameters for which the scheduler has to generate a batch of Monte-Carlo shots.

The architecture and concept of the new GREAT-ER 2 model system was designed with three important requirements: high scalability, easy extensibility and quality. The system can be used in a single user desktop environment as well as in server environments for the GREAT-ER web version. An object-oriented approach allows model developers to focus in the first instance on model components and requires less effort for data input and output and internal control flow.

4.1 Scheduler / Worker Concept

To realize a scalable model system with options for distributed computing, a two-component architecture was designed with a scheduler and a worker (Comer, 2001). The scheduler controls the simulation runs, while the worker actually performs each simulation. Each user interface, either GREAT-ER desktop or GREAT-ER web (both clients in general) is connected to one scheduler, while a scheduler can communicate with several workers. Clients do not communicate directly with a worker.

The scheduler and the worker processes communicate via the network and may run on separate machines. Workers may, of course, also run on the same system as the scheduler. Which, in turn, could run on the same machine as the client.

The clients, i.e. the GREAT-ER application server and the GREAT-ER desktop user interface, connect to the server to start and stop simulations, to query the current status of a simulation or to retrieve information about the supported models. The scheduler can manage multiple concurrent simulations for different users and from different clients.

For any given simulation the scheduler generates a batch of Monte-Carlo shot parameter sets and distributes portions to available workers. If at all possible the scheduler will only use workers that are not already working on a simulation. If all workers are already working on a simulation the scheduler will reuse one of the workers which then will have to work on two simulations at the same time.

The worker accomplishes this by calculating single complete Monte-Carlo shots for each of the simulations it handles in turn.

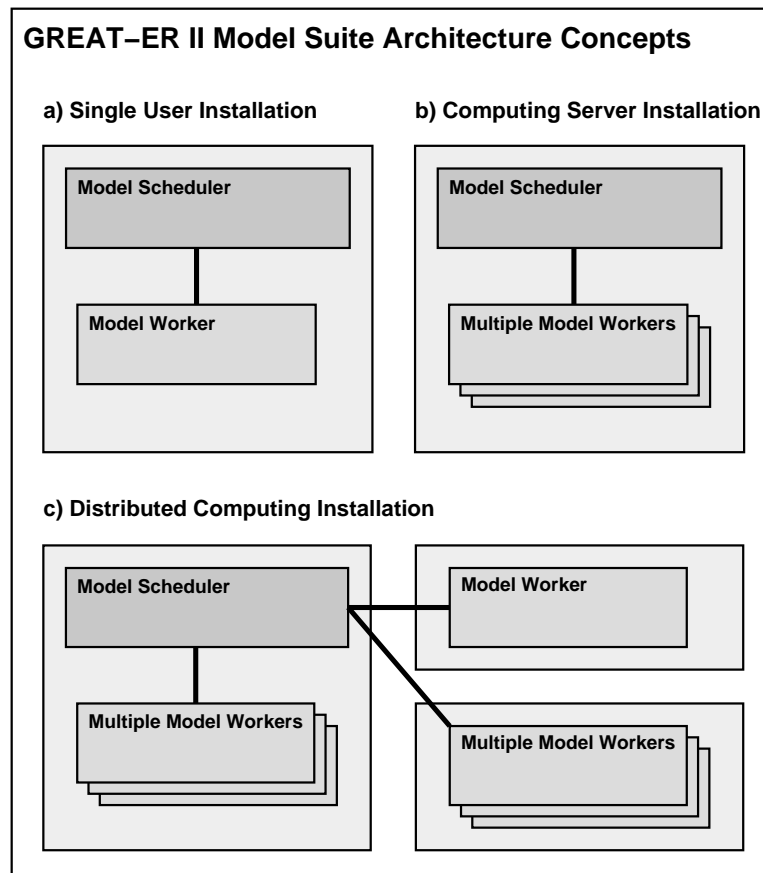


Fig. 3. GREAT-ER 2 Model System Architecture Concepts

This concept allows various combinations ranging from local installation to distributed computing, to efficiently use available computing power:

- a) Scheduler and one worker on one computer
This is the usual configuration for a single user desktop environment. When running the GREAT-ER desktop version this combination is started automatically on authentication. The model system can perform one simulation at a time. (Figure 3 a)
- b) Scheduler and several workers on one computer
The scheduler delegates complete simulation jobs to single workers, the model system can perform as many simulations in parallel as there are workers running. This setup is used for GREAT-ER web. It is also usable for network installations of GREAT-ER2, if the clients are running on low performance terminals, with the model system running on a computing server. (Figure 3 b)
- c) One scheduler and multiple workers on several computers
The scheduler splits one simulation job to different workers (running on different computers), the model system can perform at least as many simulations

in parallel as there are workers running on different computers. This installation can be used for highly frequented GREAT-ER web installations or for computing "farms" when running high numbers of Monte-Carlo shots on large catchments. (Figure 3 c)

4.2 Object-oriented Approach

The object-oriented approach (Meyer, 1988) of the GREAT-ER model system encapsulates most of the data input/output and administrative communication in a base model class. Hence a new model has to implement only model-specific methods - at least the model equations - and inherits all other methods.

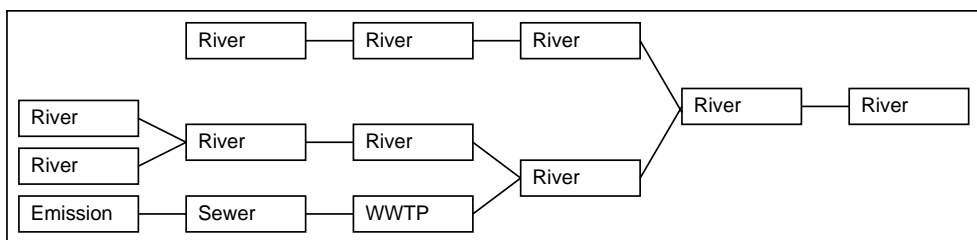


Fig. 4. GREAT-ER 2 directed graph

This object-oriented approach allows a generic function to build a directed graph (Fig. 4) representing the river network and discharge path independent from detailed model knowledge, so new models can be added easily by model developers without detailed knowledge about the internal programming.

To add a new model for a compartment already considered by GREAT-ER (emission, sewer, WWTP, river) only a couple of methods have to be defined for a new model derived from the base model class (Table 1).

5 User interfaces

Two user interfaces have been implemented in GREAT-ER 2: GREAT-ER desktop and GREAT-ER web. Both interfaces provide access to the model input parameters and the model results.

Quality assurance is guaranteed by a two-tier approach, which checks the plausibility of input data at the data entry stage. The user is warned if an unusual value is entered (although the value is accepted). An error is generated if an entered value lies outside of its valid range. Ranges are stored in the data base parameter description and can be specified when a new parameter is defined. The entire function-

Table 1
Minimum set of elements to be implemented for a new model

| Element | Description |
|-----------------------------------|--|
| <code>_required_parameters</code> | A list of parameters required by the model. The parameters are identified simply by their name and group. |
| <code>setup()</code> | The setup method is used to initialize the segment variables with the required parameters. The parameters are available from the model framework and can simply be referenced by name and group. No direct access to the database is needed. |
| <code>compute()</code> | The compute method implements the model equations which compute the segment results and the mass flux into the next segment. All values are available as set up. |

ality is implemented in a new Python module: SciParam — Scientific Parameters (Arendsen Hein, 2003).

5.1 Desktop

GREAT-ER desktop implements the full featured interface for the GREAT-ER model system. The interface is based on the platform-independent GIS data viewer Thuban (implemented in Python (van Rossum and Drake, 2003) and C (Kernighan and Ritchie, 1988) using wxWindows/wxPython (Hammond and Robinson, 2000)).

The menu items are ordered according to the usual work flow of a simulation session, similar to GREAT-ER 1.0:

- As a first step, substance has to be specified or loaded from the database.
- The selection of a catchment is the second step: For the purposes of a chemical exposure assessment, users will usually simulate a specific substance with several catchments.
- In addition to the catchment specific and detailed data (e.g. a record for each stretch in a catchment) general environmental data can be edited (default values are available).
- To run a simulation the model chain has to be assembled. Usually the modelling is an iterative process: Detailed models are used if the simple ones identify issues.
- Finally the model results can be analyzed, mathematically as well as visually (e.g. combining the results with data from monitoring)

Simulated concentrations (mean or a selected percentile) for each stretch are dis-

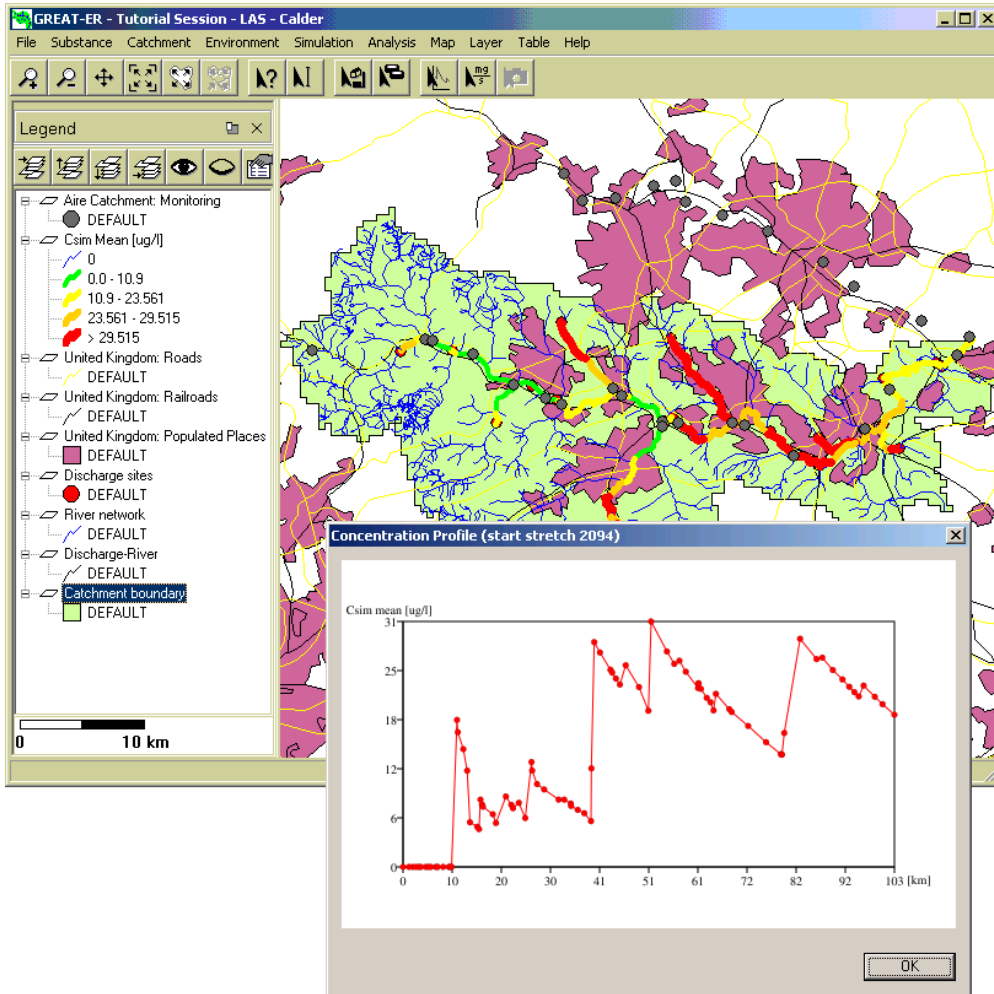


Fig. 5. GREAT-ER desktop client showing a session on the Calder catchment, UK, with a concentration profile for a selected section of the main channel inset.

played as a color coding of the river network or as a concentration profile for a selected stream of the network (Figure 5).

The three menu items Map, Layer and Table provide access to the Thuban features like map navigation, queries, table management including joins and export for external analysis.

5.2 Web

The GREAT-ER web interface provides the simplest way of access to GREAT-ER. Only a modern web browser is needed, no further installation or software licenses are required by the user. The client allows new users to get familiar with GREAT-ER concepts. The interface only offers the lowest (simplest) model mode and only provides PEC calculations and concentration profiles as analysis features. A color

coded representation of simulation results is provided (Figure 6), as well as concentration profiles downstream for a selected river stretch.

The interface is split into three frames, with a menu, the map view and a frame displaying the various dialogs. The application server managing the dialogs and concurrent sessions is implemented based on the application framework Zope (Tegtmeyer, 1999). The map display and navigation is provided by the UMN MapServer. Both components are Free Software.

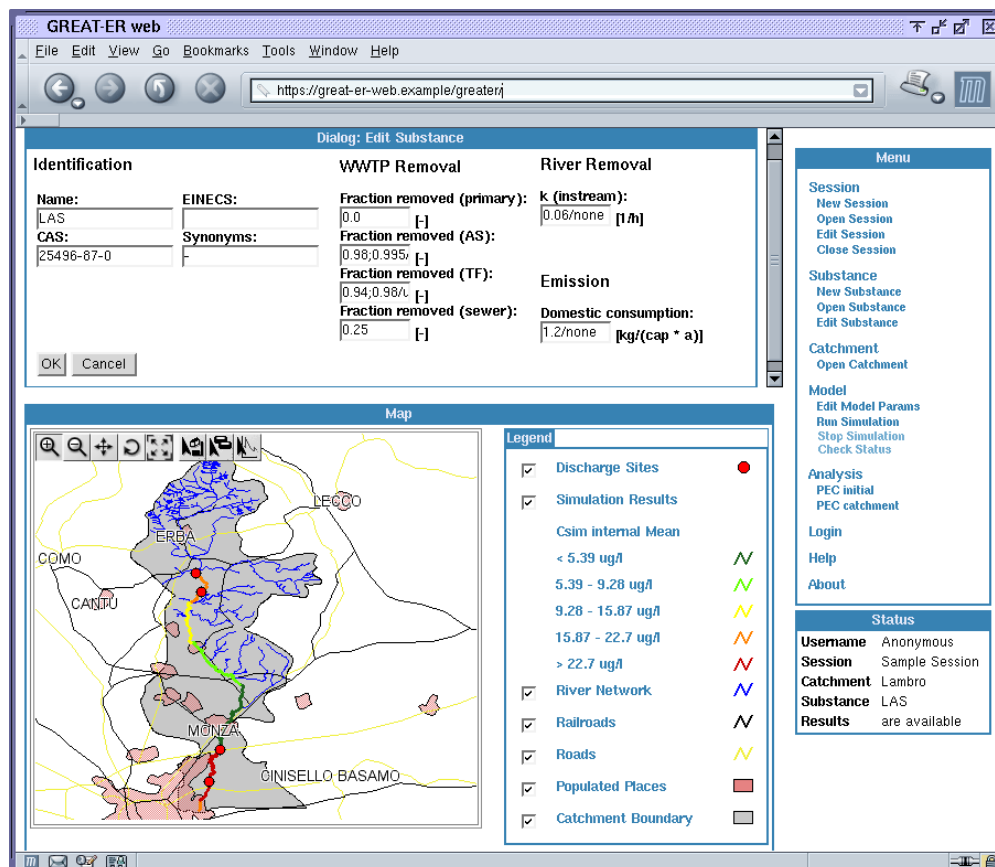


Fig. 6. GREAT-ER web client showing an implementation for the Lambro catchment, Italy

5.3 Repository

The GREAT-ER Repository is a structured collection of substance data and related documents. All this information can be stored and searched for within the database sorted according to categories.

The definition of a substance structure is not fixed and the user has the possibility to store data which are not required for the simulation but which may represent relevant background information related to the GREAT-ER data (e.g. water quality data, regulatory information, water classification). Different types of related items

(e.g. pictures, analysis of other models, laboratory data, statistics) can be stored in the document part of the GREAT-ER Repository.

The repository interface (Figure 7) is designed using wxWindows and the GREAT-ER API to access the database. The substance and document tabulator share the same principle of usage: The upper part offers several fields to enter search criteria, the middle part is designed to show the results of the search and the bottom part integrates the repository functions to edit and view data.

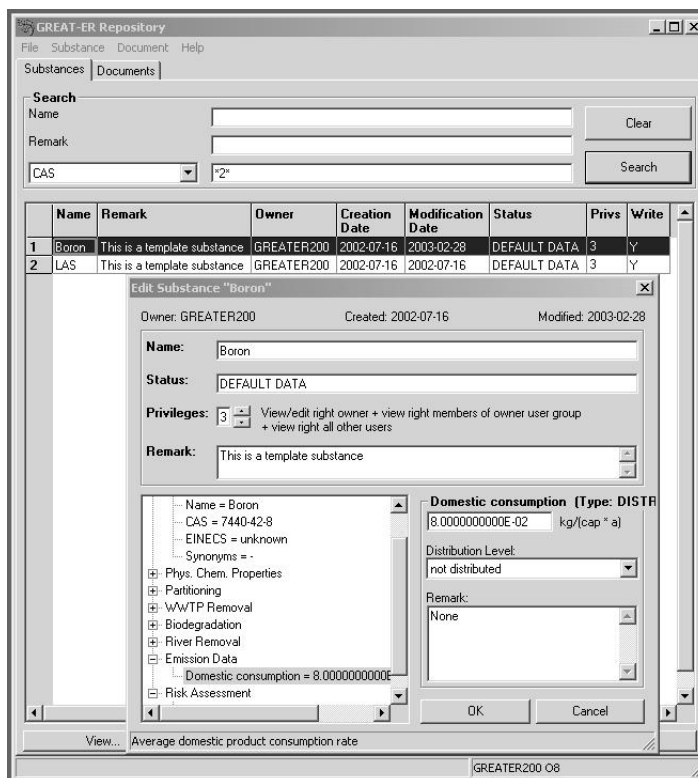


Fig. 7. GREAT-ER Repository

A web version for the Repository is also available with full functionality.

5.4 Administration

The administration interface provides access to the basic GREAT-ER elements for the experienced user. The tool (Figure 8) allows the administrator to manage users, session data, graphical binary objects, parameters and other GREAT-ER components. Also included is the XML export and import functionality that allows easy exchange of complete session and catchment data between different GREAT-ER systems. This XML interface can serve as an exchange platform to other model systems and databases.

To ensure the integrity of the system, access to the administrative functions is only

The screenshot shows the 'GREAT-ER Administration' window with a menu bar (File, Access, Customizing, Base Data, GIS Data, Exchange, Help) and a toolbar (User, Group, Parameter Tree, Phrase, Message, Session, Substance, Environment, Catchment, Binary Object, Background). The main area contains a table with the following columns: Field ID, ck, Group ID, Variable Type, Value, Field Sort, Unit, Label, Default Value, Unit, Warning Range, and Error Range. The table lists various parameters such as UP1_ID, UP2_ID, USE_MONOD, VOLATIL_MODEL, FLUX_NONDOM, FLUX_RUNOFF, H_AIR, CASNO, FLOW_DOM, EINECSNO, %101, %201, %, NAME, FLOW_NONDOM, POP, FLOW_RUNOFF, and CONSUMPTION. At the bottom, there are buttons for 'New...', 'Edit...', 'Delete', and 'Reload', and the text 'GREATER200 08' is visible in the status bar.

| Field ID | ck | Group ID | Variable Type | Value | Field Sort | Unit | Label | Default Value | Unit | Warning Range | Error Range |
|----------|----|-------------------|---------------|-------|------------|------|------------------------------|---------------|-------------------|---------------|-------------|
| 64 | | ST STRETCH_TAB | SI INT | -1 20 | 0 0 | | Upstream Stretch 1 | | | | |
| 65 | | ST STRETCH_TAB | SI INT | -1 25 | 0 0 | | Upstream Stretch 2 | | | | |
| 66 | | M SELECTION_AS | MI P42 | 3 20 | 0 0 | | Use monod | N | | | |
| 67 | | M SELECTION_RIN | MI P42 | 3 50 | 0 0 | | Volatilization | Y | | | |
| 68 | | M MARKET | M FLOAT | 3 20 | 3 1 | | Additional nondomestic input | | kg/a | | [0.00000 |
| 69 | | M MARKET | M FLOAT | 3 30 | 3 1 | | Additional runoff input | | kg/a | | [0.00000 |
| 70 | | EN ENVIRONMENT | EN FLOAT | 3 50 | 0 1 | | Air mixing height | 10 | m | | [0.00[|
| 71 | | SU IDENTIFICATION | SU CHAR | 3 20 | 0 1 | | CAS | | | | |
| 72 | | DI DISCH | DI FLOAT | 3 20 | 1 1 | | Domestic flow | | L/(cap * d) | | [0.00000 |
| 73 | | SU IDENTIFICATION | SU CHAR | 3 30 | 0 1 | | EINECS | | | | |
| 74 | | EN ENVIRONMENT | EN HEADER | 2 0 | 0 1 | | Environmental | | | | |
| 75 | | M GENERAL | MI HEADER | 2 0 | 0 1 | | General | | | | |
| 76 | | SU IDENTIFICATION | SU HEADER | 2 0 | 0 1 | | Identification | | | | |
| 77 | | SU IDENTIFICATION | SU CHAR | 3 10 | 0 1 | | Name | | | | |
| 78 | | DI DISCH | DI FLOAT | 3 30 | 1 1 | | Non-domestic flow | | m ³ /s | | [0.00000 |
| 79 | | DI DISCH | DI FLOAT | 3 10 | 1 1 | | Population | | cap | [0.000000000 | [0.00000 |
| 80 | | DI DISCH | DI FLOAT | 3 40 | 1 1 | | Runoff flow | | m ³ /s | | [0.00000 |
| 81 | | M MARKET | M FLOAT | 1 0 | 0 1 | | Site specific consumption | | kg/(cap * a) | | [0.00[|
| 82 | | SU IDENTIFICATION | SU CHAR | 3 40 | 0 1 | | Synonyms | | | | |

Fig. 8. GREAT-ER Administration: Record-based access to the data, parameter tree

possible by login as a specified GREAT-ER administration user. Parts of the functionality like changing the user password and the import/export function can also be executed by login as a normal GREAT-ER user.

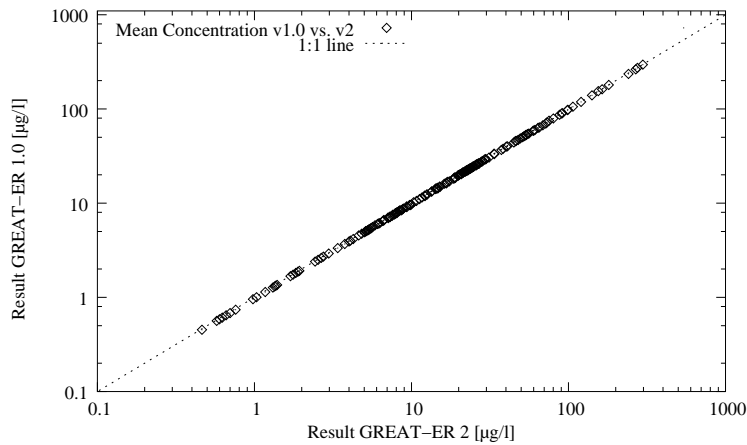
6 Comparison with GREAT-ER 1.0

For a comparison of the scientific output of GREAT-ER 2 with its predecessor GREAT-ER 1.0 a series of model runs was performed based on the Aire-Calder catchment. The Aire-Calder is a catchment in Yorkshire, United Kingdom. The catchment drains a total area of 1940 km², the digital river network consists of 3050 stretches. The model chain was setup with percentage removal for the sewer (mode 2), percentage removal for WWTP (mode 1) and combined first order elimination for rivers (mode 1). Each model run was performed with 5000 Monte-Carlo shots. The simulations were performed on a network of PCs with Intel Pentium 4, 2.2 GHz processors, running under Debian GNU/Linux 3.0.

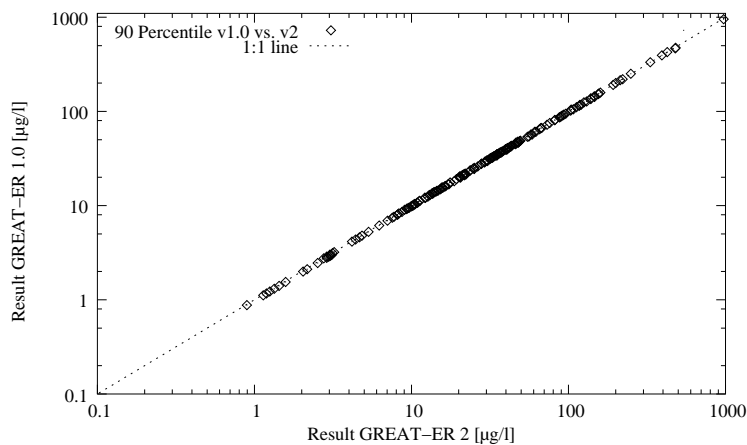
6.1 Consistency of model results

GREAT-ER 2 re-implements all models contained in GREAT-ER 1.0, using identical concepts and equations. The implementation was quality controlled using the model output produced by GREAT-ER 1.0 (which itself has been validated for a number of catchments across Europe). The quality control exercise compared the results from GREAT-ER 2 with available results from GREAT-ER 1.0.

The criterion for a successful comparison between the two models in deterministic mode (i.e. the model predictions were indistinguishable) was set at $1 \cdot 10^{-8} [mg/l]$. The criterion was successfully achieved for all stretches, suggesting that GREAT-ER 2 has been implemented successfully.



(a) Mean Concentrations



(b) 90 Percentiles

Fig. 9. Comparison of concentrations of LAS generated by GREAT-ER 2 with those from GREAT-ER 1.0 for each stretch in Monte-Carlo simulations for the Aire-Calder catchment. The quality of the underlying pseudo-random number generators and the combination of probability density functions was tested by comparing GREAT-ER 2 Monte-Carlo simulations with those from GREAT-ER 1. Figure 9 a) shows reach-specific mean concentrations of the anionic surfactant linear alkylbenzene sulphonate (LAS) generated by GREAT-ER 1.0 and GREAT-ER 2 for in Monte-Carlo simulations for the Aire-Calder catchment. The 1:1 line indicates a perfect match between the two models. The mean concentrations for each stretch generated by the two models vary between 0.05% and 2.98% with an average of 1.51%. The analysis show similar results for the comparison of 90 percentiles (Figure 9 b): For each stretch the

concentrations calculated by the two models vary between 0.01 % and 3.6 % with an average of 1.22 %. This deviation for mean values as well as for extreme values was deemed to be acceptable.

6.2 Performance

The GREAT-ER 2 Model System was designed with two major over and above those of the original GREAT-ER model: Flexibility/scalability and easy enhancement. It was expected that the new object oriented and transparent approach would result in a loss of performance compared with the optimised GREAT-ER 1.0. One strategy to overcome the expected loss has been the distributed computing approach, a second approach has been the implementation of a schedule optimization: While GREAT-ER 1.0 always simulates the entire river network, a worker of GREAT-ER 2 (Opt, i.e. optimized) can limit its model run to the currently loaded segments.

Table 2 compares the average computing times for the two model systems, including optimization and distributed computing.

Table 2

Performance Comparison of Distributed Computing, 5000 shots, Aire catchment, all timings are [m : s]

| Model System | Single User | Two Server | Three Server |
|------------------|-------------|------------|--------------|
| GREAT-ER 1.0 | 8:50 | – | – |
| GREAT-ER 2 | 16:00 | 8:46 | 6:10 |
| GREAT-ER 2 (Opt) | 6:51 | 4:18 | 3:21 |

In a plain setting GREAT-ER 1.0 is clearly faster than GREAT-ER 2, but is outperformed by the optimised version. Distributed computing increases the performance by a factor of 1.8 / 2.6 with two or three servers for GREAT-ER 2 and by a factor of 1.6 / 2.1 for GREAT-ER Opt respectively.

7 Discussion

7.1 Calculation Consistency

Deterministic predictions and the predicted mean and 90th percentile river water concentrations generated by GREAT-ER 2 compare favourably with those generated by GREAT-ER 1.0. This suggests that the model algorithms have been implemented correctly in the new software. The small differences in the determin-

istic results can be explained by the use of the different programming languages (C versus Python) with their different representation of float values and internal rounding operations. The comparison of stochastic results for both, GREAT-ER 1.0 and GREAT-ER 2 also shows good compliance, equality cannot be expected for stochastic approaches.

7.2 *Performance*

The new software was shown to be less efficient, computationally than its predecessor when used in its simplest single user mode. This is due to the fact that GREAT-ER 1.0 has been optimized by making extensive use of multidimensional arrays of values and accessing these with incremented pointers in C. This concept made it very hard to extend the model or even maintain the source and track down bugs, especially for modellers, who often are not computer scientists. GREAT-ER 2 replaces this concept with a slower but much more transparent Python dictionary (associative array). However, with respect to Moore's Law (Moore, 1965) the new implementation is already significantly faster than the performance objectives of the GREAT-ER 1.0 project.

The optimization step to limit model runs to currently loaded stretches on the other hand increases the performance significantly – as long as relevant fractions of the catchment can be considered as unloaded. This depends on the catchment structure and also on the release scenarios. In a diffuse input scenario all stretches would be loaded. The use of distributed computing to run GREAT-ER 2 also improved performance. The nonlinear factors can be explained by the administrative overhead needed for the process communication and the worker setup: Each worker has to build the directed graph representing the river network in addition to performing 2500 (two workers) or 1667 (three workers) shots. The comparison of factors for GREAT-ER 2 and GREAT-ER 2 Opt also illustrates the additional overhead for determining the loaded stretches: This task must be performed by each worker in the distributed computing setting and cannot be shared. Hence the performance increase is slightly lower than for the default implementation. On the other hand the small overhead results in a better overall performance.

7.3 *Software Architecture and Concept*

The designed software architecture and the conceptual decision to select development tools under the criteria of platform independence have proven their success: GREAT-ER 2.0 was developed and released as Free Software (under the GNU General Public License) with requirements for Microsoft® and an ORACLE® database. However, as pointed out by the performance measurements above the system also runs under GNU/Linux systems. The design and implementation of the database

API (based on Embedded SQL) enabled a porting from Oracle to the Free Software DBMS PostgreSQL. The latter development is still at an experimental stage but is available from the GREAT-ER CVS repository⁴. The database porting as well as other ongoing work by modellers to enhance GREAT-ER 2 with new models is not in the focus of this article. However, these examples illustrate that the new design and concept of GREAT-ER 2 promote further scientific work and therewith can be considered a success.

8 Conclusion

The GREAT-ER software simulates the fate of chemicals emitted to sewer systems and surface waters. It has been validated in a number of catchments in several European countries for a range of chemical types and has been used by industry and environmental regulators within the context of the environmental risk assessment of chemicals. This paper describes the development of a significant re-design of the GREAT-ER software. The new system is shown to be compatible with its well-validated predecessor in terms of simulated concentrations whilst offering significant advantages in terms of system architecture.

9 Distribution Information

GREAT-ER 2 is released as Free Software under the GNU General Public License. A CD with the desktop version of GREAT-ER 2 (requiring Microsoft® Windows 98, NT, 2000 or XP, ORACLE® database Version 8.1.7) with the complete functionality and all tools can be obtained via the GREAT-ER web site: www.great-er.org.

The GREAT-ER web version is available for public access at the same site. Without registration users have access to a limited number of catchments. Unregistered users can run a simulation but they are not able to store their settings and results permanently. Registered web users have full access to all available catchments, they can store their data.

10 Acknowledgments

The authors would like to thank the European Chemical Industry Council (CEFIC) for funding the project in the framework of the CEFIC Long-Range Research Ini-

⁴ <http://great-er.intevation.org/cvs.html>

tiative (LRI).

We would also like to gratefully acknowledge the various reports and comments from test users during the development of GREAT-ER 2, especially Andrew Riddle and his team at AstraZeneca and Bernd Scharenberg from the German Umweltbundesamt.

References

- Arendsen Hein, T., 2003. SciParam Manual, <http://sciparam.intevation.org>
- Boeije, G., Vanrolleghem, P. and Matthies, M., 1997. A Geo-referenced Aquatic Exposure Prediction Methodology for 'Down-The-Drain' Chemicals. Contribution to GREAT-ER # 3. *Wat. Sci. Technol* 36, 251-258.
- Bellasio, R. and Bianconi, R., 2005. On line simulation system for industrial accidents. *Environmental Modelling & Software*, 20, 329-342.
- Boeije, G., 1999. Chemical Fate Prediction for Use in Geo-Referenced Environmental Exposure Assessment. PhD thesis, University of Gent, Belgium.
- Boeije, G.M., Wagner, J-O., Koormann, F., Vanrolleghem, P.A., Schowanek, D.R. and Feijtel, T.C.J., 2000. New PEC definitions for river basins applicable to GIS-based environmental exposure assessment. Contribution to GREAT-ER # 8. *Chemosphere*, 40, 255-265.
- Boos S., 2001. IUCLID 4.0 Reference Manual
http://www.technidata.de/iuclid_doc
- Comer, D.E., 2001. *Internetworking with TCP/IP, Volume 3: Client-Server Programming and Applications Linux/Posix Sockets Version*. Prentice Hall
- European Commission, 2003. Technical Guidance Documents in Support of the Commission Directive 93/67/EEC on Risk Assessment of new Substances and Commission Regulation (EC) No. 1488/94 on Risk assessment for existing Substances. Brussels, Belgium.
- Feijtel, T.C.J., Boeije, G., Matthies, M., Young, A., Morris, G., Gandolfi, C., Hansen, B., Fox, K., Holt, M., Koch, V., Schröder, R., Cassani, G., Schowanek, D., Rosenblom, J. and Niessen, H., 1997. Development of a Geography-referenced Regional Exposure Assessment Tool for European Rivers - GREAT-ER. Contribution to GREAT-ER # 1. *Chemosphere* 34, 2351-2373.
- Gandolfi, C., Facchi, A., Whelan, M.J., Cassani, G., Tartari, G. and Marcomini, A., 2000. Validation of the GREAT-ER model in the River Lambro catchment. Proceedings of 5th CESIO World Surfactant Congress Fiorenza May 29-31.
- Halls, J. N., 2003. River run: An interactive GIS and dynamic graphing website for decision support and exploratory data analysis of water quality parameters of the lower Cape Fear river. *Environmental Modelling & Software*, 18, 513-520.
- Hammond, M., Robinson, A., 2000. *Python Programming on Win32*. O'Reilly
- Holt, M.S. Daniel, M., Buckland, H. and Fox, K.K., 2000. Monitoring studies in the UK designed for the validation of the geo-referenced exposure assessment tool

- for European rivers (GREAT-ER). Proceedings of 5th CESIO World Surfactant Congress Fiorenza May 29-31.
- Kernighan, B.W., Ritchie, D.M., 1988, The C Programming Language, Second Edition, ANSI C, Prentice Hall International
- Koormann, F., Matthies, M. and Trapp, S., 1998. Simulation and Visualisation of Spatial Exposure Patterns: Intermediates in the Rhine River System. In: Regulation for Chemical Safety in Europe: Analysis, Comment and Criticism (Eds. D.M. Pugh, J.V. Tarazona), Kluwer Academic Publishers, Dordrecht, pp. 65 - 96
- Matthies, M., Wagner, J.-O. and Koormann, F., 1997. Combination of Regional Exposure Models for European Rivers with GIS Information. In: ECO-INFORMA 12: Information and Communication in Environmental and Health Issues, 523 - 529
- Meyer, B., 1988. Object-Oriented Software Construction. Prentice Hall
- Moore, G. E., 1965. Cramming more components onto integrated circuits Electronics, Volume 38, Number 8, April 19, 1965
- van Rossum, G., Drake, F., 2003. Python Language Reference Manual. Network Theory Ltd.
- Schowaneck, D., Fox, K., Holt, M., Schroeder, R., Koch, V., Cassani, G., Matthies, M., Boeije, G., Vanrolleghem, P., Young, A., Morris, G., Gandolfi, C. and Feijtel, T., 2001. GREAT-ER: A new tool for management and risk assessment of chemicals in river basins. Contribution to GREAT-ER # 10. Wat. Sci. Technol. 43, 179-185.
- Schulze, C., Matthies, M., 2001. Georeferenced aquatic fate simulation of cleaning agent and detergent ingredients in the river Rur catchment (Germany). Sci. Tot. Environ. 280, 55-77
- SEPA, 2002. Modelling Continuous Discharges to Rivers: Technical Guidance Manual for Licensing Discharges to Water, Scottish Environment Protection Agency, Stirling, Scotland 41pp
- Shena, J., Parker, A. and Riverson, J., 2005. A new approach for a Windows-based watershed modeling system based on a database-supporting architecture. Environmental Modelling & Software, 20, 1127-1138.
- Trapp, S., Matthies, M., 1998. Chemodynamics and environmental modeling: An introduction. Springer Verlag, Berlin
- Tegtmeyer, F., 1999. Objekthierarchie (in German). iX, Heise Verlag, Hanover, 64.
- Verdonck, F., Boeije, G., Vanrolleghem, P.A., Feijtel, T.C.J. and Schowaneck, D., 2000. Toepassing van GREAT-ER, een blootstellingsmodel voor chemicalien, op het Rupelstroombekken (in Dutch). WEL 4-2.
- Wagner, J.-O., 2001. Spatial Aspects of Chemical Exposure Assessment: A Tool for River Networks. Networks. Dissertation an der Universität Osnabrück.
http://elib.ub.uni-osnabrueck.de/publications/diss/E-Diss154_thesis.pdf
- Warn, A.E. and Brew, J.S., 1980. Mass Balance. Water Research, 14, 1427-
- Warn, A.E. 1982. Calculating consent conditions to achieve River Quality Consent Objectives. Effluent and Water Treatment Journal, 22, 152-155