DIPLOMARBEIT

Quality Assurance of Emission Data in the RAINS-Asia Model

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines
Dipl.–Ing. (FH) für Computersimulation
am Fachhochschul-Diplomstudiengang Computersimulation St. Pölten

unter der Leitung von:

Dipl.–Ing. Dr. Wolfgang Schöpp

Dipl.–Ing. Dr. Christian Harlander

ausgeführt von

Robert Sandler
si021020

St. Pölten, September 2006

DIPLOMA THESIS

Quality Assurance of Emission Data in the RAINS-Asia Model

for the purpose of acquiring the academic degree

Dipl.–Ing. (FH) of Computersimulation

at the University of Applied Science St. Pölten

under the guidance of :

Dipl.–Ing. Dr. Wolfgang Schöpp

Dipl.–Ing. Dr. Christian Harlander

proposed by

Robert Sandler

si021020

St. Pölten, September 2006

Laxenburg, 1. September 2006

I, Robert W. Sandler hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to any substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the diploma thesis.

. . . .

Robert Wolfgang Sandler

Zusammenfassung

In vielen Ländern Asien boomt die Wirtschaft und dieser Trend scheint sich, in den nächsten Jahren fortzusetzen. Der Wachstum geht Hand in Hand mit einer Steigerung des Engergieverbrauches. Da die grösste Energiequelle Kohle ist, wird die Luft auch in gleicher Massen verschmutzt. Um effizientere Kontrollstrategien zu entwickelen, plant IIASA eine Erweiterung bzw. eine Einbindung von Südostasien in das Rains Web Model (RAINS-Asia). Dies bedeutet ein erhöhtes Datenvolumen. Rains-Asia wird aus 95 Emissionsregionen in 23 Ländern bestehen. Zum Vergleich, ein Szenario für Europa mit 50 Ländern benötigt die Ressourcen von über einer Million Datensätze. Um diese Anzahl bestmöglichst bearbeiten und kontrollieren zu können, müssen Kontrollstrategien entwickelt werden, die höchste Qualität der Daten gewährleisten.

RAINS WEB (Browser Simulation Tool basierend auf dem RAINS Modell) ist ein Multiuser Model. Eine priviligierte Gruppe von Benutzern ist ermächtigt Daten über das Internet zu modifizieren und in das bestehende System zu laden. Um alle Änderungen am System im Auge zu behalten, und nachvollziehen zu können, werden Metadaten im System mitgespeichert. Die Daten über den User und die vorgenommenen Änderungen sollen mit Datum und Uhrzeit in transparenten Reports zur Verfügung gestellt werden. Zusätzlich sollen Fehlermeldungen über fehlgeschlagene Datenuploads entwickelt werden. Einfache Konsistenzchecks (z. B. ob die Parameter eine gewisse Grenze überschreiten, Summen unter 100 %, nicht negativer Wert, ...) müssen entwickelt werden. Mit Hilfe dieser Erweiterungen soll es möglich sein, die Datenqualität zu gewährleisten und interne Fehler auszuschliessen. Auch auf Grund der hohen Datenmenge sollen diese Tests und Checks automatisiert werden. Abschliessend stehen die Ergebnisse dieser Test dem Supervisor zur Verfügung, welcher dann letztendlich die Daten freigibt damit die Daten in der Datenbank gespeichert werden können.

Abstract

Many Asian countries have experienced rapid economic growth during recent years and this trend is expected to continue. The economic growth is accompanied by increasing energy demand, with coal as the dominant energy source. A significant increase in emissions of various air pollutants is therefore foreseen. In order to develop efficient control strategies for that part of the world, IIASA plans an extension of the RAINS WEB model to southeast Asia. This will substantially increase the data volume to be handled. RAINS-Asia will include 95 emission regions in 23 countries. For comparison, one scenario for Europe, covering 50 countries, requires preparation and storing of about one million datasets. Management of such a large number of data sets requires development of tools and practices to ensure the quality of data.

RAINS WEB (browser simulation tool based on the RAINS model) is a multi-user model. A ”privileged” group of users has a right to modify the input data and upload the modifications via the Internet. To keep track of all the modifications it is planned to develop a set of metadata to be stored during the upload. In particular, information about who has uploaded what and when needs to be stored and presented in a form of transparent reports. Besides, a diagnosis of formal errors and the reasons for upload failure needs to be developed. Simple consistency checks (e. g., whether the uploaded parameter is within the allowed range, control sums of other parameters, calculation of indicators and checking their order of magnitude) need to be developed. This will enable the IIASA staff to spot the internal inconsistencies of data and prevent the use of inconsistent. Because of the large amount of data to be handled, the consistency checks need to be automated. Before the data is used in the environmental assessment it will need to be approved by a responsible person. The system will help to administer the approval procedure.

Contents

Dankesworte

Seitens des Institutes für angewandte Systemanalyse (IIASA) möchte ich mich beim gesamten Team der Abteilung APD (Atmospheric Pollution and Economic Development) für das herzliche Willkommen bedanken. Es ist nicht selbstverständlich so schnell in das Team integriert und akzeptiert zu werden. Bezüglich meiner Arbeit möchte ich mich im Besonderen bei Hr. Adam Chambers, Hr. Chris Heyes und Hr. Dr. Fabian Wagner für die Durchsicht meiner Diplomarbeit bedanken. Besonderer Dank gilt meinem Supervisor Hr. Dr. Wolfgang Schöpp für die Hilfe und Anregungen bei meiner Arbeit.

Seitens der Fachhochschule St. Pölten möchte ich Hr. Dr. Christian Harlander für die Durchsicht, Anregungen und Hilfe bei meiner Arbeit danken.

Bei meiner Familie möchte ich mich für die Unterstützung während meiner Studienzeit bedanken. Besonders bei meiner Mutter die ich, umso näher das Studienende kam, immer öfter vertrösten musste und der Kontakt nur mehr telefonisch stattfand. Bedanken möchte ich mich auch bei meinem Vater, der es immer verstand mich aufzubauen. Meinem Bruder Gregor danke ich, dass er immer zu mir gehalten hat und seinen grossen Bruder mit Stolz erfüllt.

Der Mutter meiner Freundin, Elisabeth Strummer, möchte ich danken, dass Sie mich auf eine gewisse Fachhochschule in St. Pölten aufmerksam gemacht hat und mir während der Studienzeit viele Lasten von den Schultern genommen hat.

Last but not least möchte ich meiner Freundin Isabella danken, die mein Ruhepol in schwierigen Zeiten war und es auch immer verstand mir zu helfen wenn es mir an was fehlte. Speziell zu Studienende war Sie eine große Stütze und hat mir Kraft gegeben.

Danke euer ROBERT

Table of acronyms

Acronym	Description
ADB	Asian Development Bank
APD	The Atmospheric Pollution and Economic Development program aims to employ IIASA's expertise in applied interdisciplinary research to develop innovative modeling tools to identify strategies to protect the local, regional and global atmosphere while imposing least burden on the economic development.
CAFE	Clear Air For Europe
CCE	Coordination Center for Effects
CLRTAP	Convention on Long Range Transboundary Air Pollution
EBRD	European Bank for Reconstruction and Development
EEA	European Environmetal Agency
EMEP	EMEP is a scientifically based and policy driven program under the Convention on Long-range Transboundary Air Pollution for international co-operation to solve transboundary air pollution problems.
GAINS	Greenhouse Gas and Air pollution INteractions and Synergies
IIASA	International Institute For Applied System Analysis
LRTAP	Convention on Long Range Transboundary Air Pollution
NMO	National Member Organizations
POI	Apache library for Excel^{TM} handling with JAVA
RAINS	Regional Air Pollution Information and Simulation
RIVM	National Institute for Public Health and Environmental Protection
UN	United Nations
UN-ECE	United Nations Economic Commission for Europe

Chapter 1

Introduction

The past decades has seen extraordinary shifts in the ways that European countries negotiate efforts to limit air pollution. Computer modeling, once a peripheral aid, has moved to the very heart of negotiations. This has radically altered the way governments think about air pollution problems and the way that they search for solutions. Modern efforts to control air pollution in Europe began in the 1970s, prompted by concerns over acid rain. British scientist R.A. Smith first warned of the problem in 1872. A century later, the acidity of precipitation in Europe had increased at least ten-fold. Studies in the 1970s confirmed that pollutants often traveled long distances before falling to earth. In 1979 the Convention on Long-range Transboundary Air Pollution was signed by 35 countries, including all states in Europe, the USA, and Canada. The convention was negotiated through the UN Economic Commission for Europe (UN-ECE), a Geneva-based body that was then one of the few international organizations that brought together European countries from East and West. The UN-ECE convention would become a framework for subsequent efforts to limit air pollution.

Initial efforts focused on acid rain. In 1985, 20 parties to the convention signed a protocol stating that by 1993 they would reduce annual emissions and "exports"of sulfur dioxide, the main source of acidification, by at least 30 percent, compared to 1980 levels.

A uniform percentage cut, while better than nothing, is crude and inefficient. It ignores that some ecosystems are very sensitive while others are not; if the goal is to protect the environment, it makes little sense to cut emissions if they fall in places where they do no harm. Moreover, across-the-board cuts do not no take into account that some emissions can be cut more cheaply and quickly than others.

The **R**egional **A**ir Pollution **IN**formation and **S**imulation (RAINS) model was designed to address these issues. In 1989, when the sulfur protocol was due for renegotiation, a task force of the UN-ECE noted that:

"An integrated assessment model that can assist in cost-effectiveness analysis is now available. ... [This task force] recommends that the RAINS model be used by Parties to the Convention, the Executive Body, and various subsidiary bodies."

This was an historic resolution. For the first time, all parties to a major international negotiation accepted one computer model and agreed to make it a key tool in their negotiations. The nearest parallel occurred during the 1970s negotiation of the Law of the Sea; some negotiating teams from large countries allowed smaller delegations to use their simulation models, but this was done informally. RAINS, by contrast, would play a central role in renegotiation of the sulfur protocol. [RAINS Options 1998]

1.1 Simulation

A simulation is an imitation of some real thing, state of affairs, or process [SIMDEF]. The act of simulating something generally entails representing certain key characteristics or behaviors of a selected physical or abstract system.

> ...for Distinction Sake, a Deceiving by Words, is commonly called a Lye, and a Deceiving by Actions, Gestures, or Behavior, is called Simulation ...
>
> Robert South (1643-1716)

A computer simulation is an attempt to model a real-life situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behavior of the system.

Simulation is used in many contexts, including the modeling of natural systems or human systems in order to gain insight into their complexity. Other contexts include simulation of technology for performance optimization, safety engineering, testing, training and education. Simulation can be used to show the eventual real effects of alternative conditions and courses of action.

Key issues in simulation include acquisition of valid source information about the referent, selection of key characteristics and behaviors, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes.

Computer simulation has become a useful part of modeling many natural systems in physics, chemistry and biology, and human systems in economics and social science (the computational sociology) as well as in engineering to gain insight into the operation of those systems. Another good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations the model behavior will change each simulation according to the set of initial parameters assumed for the environment. Computer simulations are often considered to be human out of the loop simulations.

Traditionally, the formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions to problems. This enables the prediction of the behavior of the system from a set of parameters and initial conditions. Computer simulation is often used as an adjunct to, or substitution for, modeling systems for which simple closed form analytic solutions are not possible. There are many different types of computer simulation, the common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states of the model would be prohibitive or impossible. Several software packages exist for running computer-based simulation modeling that makes the modeling almost effortless and simple (e. g., Monte Carlo simulation and stochastic modeling with risk simulaton).

It is increasingly common to hear simulations of many kinds referred to as ”synthetic environments”. This label has been adopted to broaden the definition of ”simulation”to encompass virtually any computer-based representation.

1.1.1 The objectives of emission and control cost calculations

One of the central objectives of integrated assessment models is to assist in the cost-effective allocation of emission reduction measures across various pollutants, several countries and different economic sectors [Amann et. al. 2004]. Obviously, this task requires consistent information about the costs of emission factors at the individual sources, and it is the central objective to provide such information. The optimal allocation of emission control measures between countries is crucially influenced by differences in emission control costs for the individual emission sources. It is therefore of utmost importance to identify systematically the factors leading to differences in emission control costs among countries, economic sectors and pollutants. Such differences are usually caused, inter alia, by variations in the composition of the various emission sources, the state of technological development and the extent to which emission control measures are already applied.

1.2 IIASA

The International Institute for Applied Systems Analysis (IIASA) is a non-governmental research organization located near Vienna, Austria. The institute conducts inter-disciplinary scientific studies on environmental, economic, technological and social issues in the context of human dimensions of global change. It is sponsored by its national member organizations in Africa, Asia, Europe, and North America.

IIASA researchers study environmental, economic, technological, and social developments. In doing so, they generate methods and tools useful to both decision makers and the scientific community. The work is based on original state-of-the-art methodology and analytical approaches and links a variety of natural and social science disciplines.

Since IIASA’s inception in 1972, it has been the site of successful international scientific collaboration in addressing areas of concern for all advanced societies, such as energy, water, environment, risk and human settlement.

Now, after over three decades of valuable contributions to science and policy, IIASA continues as a well reputed center for innovative research, international interdisciplinary collaboration, conferences and workshops, and networking among scientists around the world.

Because of its non-governmental status, IIASA is independent and can provide non-political and unbiased perspectives. This neutrality and impartiality is particularly valued by those who utilize the institute's research findings.

IIASA is sponsored by scientific National Member Organizations (NMO's) in nations from Africa, Asia, Europe, and North America. Each NMO nominates one representative to IIASA's executive council, which generally oversees the institute's development.

1.3 Atmospheric Pollution and Economic Development (APD)

Recent scientific work indicates potentially important economic and environmental synergies between air pollution control and the mitigation of greenhouse gases at different temporal and spatial scales. A systematic assessment of such synergies and their interactions with economic development could thus point the way towards effective and viable approaches for protecting the local and global atmosphere.

The Atmospheric Pollution and Economic Development program aims to employ IIASA's expertise in applied interdisciplinary research to develop innovative modeling tools to identify strategies to protect the local, regional and global atmosphere while imposing least burden on the economic development. IIASA's work will bring together geophysical and economic aspects of pollution control into one assessment framework and implement it together with a network of collaborators for practical policy analysis in different regions of the world.

1.4 RAINS model

> ”RAINS is a scenario-generating device that helps users to understand the impacts of future actions – or inaction – and to design strategies to achieve long-term environmental goals at the lowest possible cost.”

IIASA Options [IIASAOptions]

The **R**egional **A**ir Pollution **IN**formation and **S**imulation (RAINS)-model developed at the International Institute for Applied Systems Analysis (IIASA, Laxenburg, Austria) provides a consistent framework for the analysis of emission reduction strategies, focusing on acidification, eutrophication and tropospheric ozone. RAINS comprises modules for emission generation (with databases on current and future economic activities, energy consumption levels, fuel characteristics, etc.), for emission control options and costs, for atmospheric dispersion of pollutants and for environmental sensitivities (i.e., databases on critical loads). In order to create a consistent and comprehensive picture of the options for simultaneously addressing the three environmental problems (acidification, eutrophication and tropospheric ozone), the model considers emissions of sulfur dioxide (SO_2), nitrogen oxides (NO_X), ammonia (NH_3) and volatile organic compounds (VOC). Consequently, the search for cost-effective solutions to control the ambient levels of fine particles should balance emission controls over the sources of primary emissions as well as over the precursors of secondary aerosols. Thus, the control problem can be seen as an extension of the "multi-pollutant / multi-effect"concept applied for acidification, eutrophication and groundlevel ozone (Table 1.1).

A detailed description of the RAINS model can be found in [Alcamo et. al. 1990]. A schematic diagram of the RAINS model is displayed in Figure 1.1.

	Primary PM	SO_2	NO_X	VOC	NH_3
Health impacts:					
PM	x	x	x	x	x
O_3			x	x	
Vegetation impacts:					
O3			x	x	
Acidification		x	x		x
Eutrophication			x		x

Table 1.1: Multi-pollutant / multi-effect approach of the RAINS model

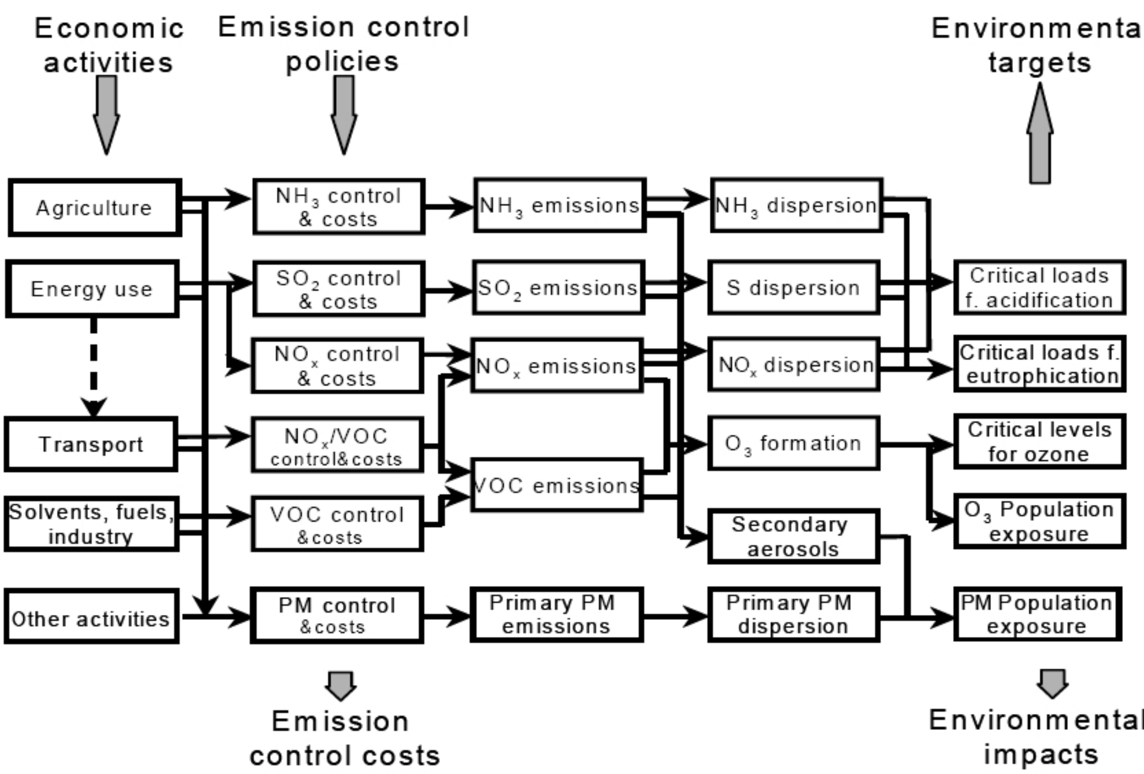

Figure 1.1: Flow chart of the RAINS model

The European implementation of the RAINS model incorporates databases on energy consumption for 40 regions in Europe, distinguishing 22 categories of fuel use in six economic sectors. The time horizon extends from the year 1990 up to the year 2010 [Bertok et. al. 1993]. Emissions of SO_2, NO_X, NH_3 and VOC for 1990 are estimated based on information collected by the CORINAIR'90 inventory of the European Environmental Agency [EEA European-EnvironmentalAgency] and on national information. Options and costs for controlling emissions of the various substances are represented in the model by considering the characteristic technical and economic features of the most important emission reduction options and technologies. Atmospheric dispersion processes over Europe for sulfur and nitrogen compounds are modeled based on results of the European EMEP model developed at the Norwegian Meteorological Institut [Barret and Sandnes 1996]. For tropospheric ozone, source-receptor relationships between the precursor emissions and the regional ozone concentrations are derived from the EMEP photooxidants model [Simpson 1993]. The RAINS model incorporates databases on critical loads and critical levels compiled at the Coordination Center for Effects (CCE) at the National Institute for Public Health and Environmental Protection (RIVM) in the Netherlands [Posch et. al. 1997]. The RAINS model can be operated in the scenario analysis mode, i.e., following the pathways of the emissions from their sources to their environmental impacts. In this case the model provides estimates of regional costs and environmental benefits of alternative emission control strategies. Alternatively, a (linear programming) optimization mode is available for the acidification part to identify cost-optimal allocations of emission reductions in order to achieve specified deposition targets. This mode of the RAINS model was used extensively during the negotiation process of the Second Sulfur Protocol under the Convention on Long-Range Transboundary Air Pollution (LRTAP) for elaborating effect-based emission control strategies. A non-linear optimization module for tropospheric ozone has been recently completed (including PM with additional secondary aerosols).

1.5 GAINS model

The **G**reenhouse Gas and **A**ir Pollution **IN**teractions and **S**ynergies (GAINS) project brings together a multinational team of experts who will develop a state-of-the-art disciplinary model to assess the co-benefits of air pollution and greenhouse gas emission reductions. Over a two-year timeframe, the team will assess the technical and market-based policies that maximize synergies within the co-benefits policy arena. GAINS-Asia will integrate policy-relevant information from the following models:

- integrated
    - BernCC carbon cycle model
    - MESSAGE global energy scenario model
    - RAINS air pollution integrated assessment model

- used
  - GAINS Model, which extends RAINS to address mitigation potentials for greenhouse gas emissions
- using inputs
  - TM5 hemispheric atmospheric chemistry and transport model
  - MARKAL and IPAC energy models for India and China respectively

GAINS-Asia will construct and validate reduced-form representations of these models and combine these functional relationships at the meta-level for the newly developed GAINS-Asia policy assessment framework. The GAINS-Asia tool will allow the interactive assessment of the cost-effectiveness and benefits analysis for a wide range of technical and market-based policy options. Optimization approaches will be developed to identify combinations of policies aimed at reducing long-range and hemispheric air pollution alongside greenhouse gas emissions in order to optimize overall benefits in the medium- and long-term. GAINS-Asia will focus specifically on medium-term policy measures for European and Asian countries that maximize synergies between air pollution control and greenhouse gas mitigation, while embedding these policy measures in global strategies that would achieve the stabilization of greenhouse gas concentrations in the long-term. In a first attempt GAINS will be implemented for China, India, and 43 European countries including Russia. To enable the analysis in a global context, the rest of the world will be represented at an aggregated level and may be updated with future additions to the GAINS/RAINS Database. An interactive web-based software will be developed to provide stakeholders with access to the GAINS-Asia Model. This web-based model will allow developing country partners to conduct independent analysis of the interactions between air pollution and climate change policies. [Wagner et. al. 2005]

1.6 Problem formulation

RAINS WEB is a multi-user model. A ”privileged”group of users has a right to modify the input data and upload the modifications via the Internet. To keep track of all the modifications a set of metadata needs to be stored during the upload. Metadata are all user depending data (IP address, username, time, ...). With the storage of the user data a historical grow of the data is being build automatically. This historical feature is needed to provide the opportunity for a possible reconstruction of the data.

In particular, information about who has uploaded what and when needs to be stored and presented in a form of transparent reports. Besides, a diagnosis of formal errors and the reasons for upload failure needs to be developed. Simple consistency checks (e. g., whether the uploaded parameter is within the allowed range, control sums of other parameters, calculation of indicators and checking their order of magnitude) need to be developed. This will allow the IIASA staff to spot the internal inconsistencies of data and prevent the use of inconsistent data.

Before the data is used in the environmental assessment it will need to be approved by a responsible person. To shorten the time of personal checking (one scenario can included 2 000 000 datasets), the consistency checks need to be automated.

The system will help to administer the approval procedure. System includes message tools (popup windows, emails, ...) will help the administrator to validated the data in more common and more comfortable way. Also the user will profit of the messages of missing data or any other kind of possible user side errors (missing values, wrong check sums, ...).

Baseline scenarios are being build out of the results of the visiting countries. The industry (e. g. printing industry) and countries are visiting IIASA and control the calculated and simulated data.

UN and EU protocols can help building baseline scenarios or even give helpful suggestions on limits and guidelines for such baseline scenarios. The Kyoto protocol (UN CO_2 directive [Kyoto]) to the United Nations Framework Convention on Climate Change (UNFCCC) was adopted on the 11th of December 1997. This treaty, which contains legally binding quantitative commitments for industrialised countries, only entered into force on 16 February 2005.

Chapter 2

Current knowledge

The RAINS-Asia model will become operational within the next few months. This project will help the World Bank [WORLDBANK], the Asian Development Bank [ADB] and the European Bank for Reconstruction and Development [EBRD] to make decisions on credits and investments on Asian projects.

Many Asian countries have experienced rapid economic growth during recent years and this trend is expected to continue. The economic growth is accompanied by increasing energy demand, with coal being the dominant energy source. A significant increase in emissions of various air pollutants is therefore foreseen.

As there are no kind of directives for such control system for air pollution, common directives for Europe used to be used for a baseline for the ASIAN version of the RAINS model.

A recent assessment of the transboundary air pollution in the European Union [LRTAP 2002] revealed that major emissions reductions for SO_2 and NO_X as adopted under the Convention on Long Range Transboundary Air Pollution (CLRTAP) and EU legislation, primarily the Directive 2001/81/EC on National Emissions Ceiling, have reduced the harmful effects associated with the presence of these substances, namely their contribution to the formation of photo chemical smog and the acidification and eutrophication (Eutrophication is the enrichment of an ecosystem with chemical nutrients, typically compounds containing nitrogen or phosphorus.) of water and soil. The multiple role of SO_2 and NO_X causes interconnection between environmental problems that does not allow a complete address of the impacts of the transboundary air pollution. [Moussiopoulus et. al. 2004]

Long-range transport of air pollutants is one of the main issues that European Union includes in its legislation. In particular, transboundary air pollution, i. e. the air pollution generated in one country and being transported to its neighbor countries is considered as a major European problem.

In order to develop efficient control strategies for that part of the world, IIASA plans an extension of the RAINS WEB model to southeast Asia. This will substantially increase the data volume to be handled. RAINS-Asia will include 95 emission regions in 23 countries. For comparison, one scenario for Europe, covering 50 countries, requires preparation and storing of about one million datasets. Management of such a large number of data sets requires development of tools and practices to ensure the quality of data.

2.1 Historical background

RAINS was the first computer model ever placed at the center of negotiations for a major environmental agreement: the 1994 agreement by 35 countries to limit sulfur dioxide emissions in Europe. Subsequently, RAINS played a central role in the creation of the European Commission's 1995 Acidification Strategy. More recently it has supported the development of new, legally binding EU directives on air quality and emissions; those directives should be published late in 1998. And, as mentioned above, RAINS is central to joint efforts by EU member countries and the nations of Central and Eastern Europe to sign a truly comprehensive agreement covering four major pollutants.

This role–helping governments to identify scientifically sound, cost-effective policies to combat air pollution–precisely fulfills the task envisaged for RAINS by the IIASA scientists who conceived the model in 1983.

The past decade has seen extraordinary shifts in the ways that European countries negotiate efforts to limit air pollution. Computer modeling, once a peripheral aid, has moved to the very heart of negotiations. This has radically altered the way governments think about air pollution problems and the way that they search for solutions.

Studies in the 1970s confirmed that pollutants often traveled long distances before falling to earth. In 1979 the Convention on Long-range Transboundary Air Pollution was signed by 35 countries, including all states in Europe, the USA, and Canada. The convention was negotiated through the UN Economic Commission for Europe, a Geneva-based body that was then one of the few international organizations that brought together European countries from East and West. The UN-ECE convention would become a framework for subsequent efforts to limit air pollution. [MacDonald 1998]

2.2 Implementation

The RAINS Web Model is a Client-Server model. A Client-Server model is a network architecture which separates the client (often an application that uses a graphical user interface) from the server. Each instance of the client software can send requests to a server or application server. Although this idea is applied in a variety of ways, on many different kinds of applications, the easiest example to visualize is the current use of web pages on the Internet.

2.3 Technical knowledge

As stated in 2.1 RAINS was developed as an stand alone Version (in C++). The needed usage as a multi user model requires a database. The database is the major task of the model. IIASA decided to use an Oracle database because Oracle provides an additional free version Oracle XE (Express Edition)[ORACLEXE]. There is also the possibility to change the database to the similar and free database *PostGreSQL*. The modification from a stand alone version to a multi user Version (Client-Server) was implemented by the language JAVA, because JAVA provides a numerous of advantages (e. g. security, stability, numerous of libraries, . . .).

Client-Server is a network architecture which separates the client (often an application that uses a graphical user interface) from the server. Each instance of the client software can send requests to a server or application server.

Client		Server
request	$\Longrightarrow$	
		process
	$\Longleftarrow$	answer

Table 2.1: Client-Server

The system is fully built in JAVA. The communication (Client $\longrightarrow$ Server) will be performed with Servlets. Each functionality is based on such a Servlet. The JAVA Servlet API allows the software developer to add dynamic content to a Web server using the JAVA platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the JAVA counterpart to dynamic web content technologies such as CGI, PHP or ASP. Servlets can maintain state across many server transactions by using HTTP cookies, session variables or URL rewriting. More information about JAVA Servlets can be found at *http://java.sun.com*.

2.4 Data transfer

The data transfer is the major task of the quality assurance. Data transfer means the up- and download of data in templates. On the one hand data is being brought to IIASA and on the other hand back to the user.

The user will have to fill in his data into given templates. The use of templates will ensure the data transfer in a given fixed format. With a fixed format, checks can be implemented in a easy way. For example: If the user failed to fill in all data the templates can be prepared with pre-checks that shows up such errors (missing data, exceeding range, . . .).

The data values are the exchanged information of the ”communciation”. That‘s why spreadsheets[1] are the best baseline for the templates.

[1]A spreadsheet is a rectangular table or grid of information, often financial information

Spreadsheets will build the base of data transmission. Any type of spreadsheet (MS Excel^{TM}, OpenOffice, ...) can be used. It has been decided that templates prepared in MS Excel^{TM} are the platform for data exchange. Such templates have a fixed, well-defined format that users have to follow. Fixed format means fixed rules that can be checked. As indicated later 3.1.2 testing procedures will be developed.

Microsoft Excel^{TM} is the world's leading spreadsheet program [Maran 2003], used and supported in over 40 countries. After Word, it is the second most used module in the Microsoft Office suite.

The communication (data exchange) with the users will be performed on strict, nested templates. The first level will provide principal data and parameters while the second, more accurate, will comprise all data and parameters used in the calculation. Depending on the purpose of the project and experience of the users the respective template can be used.

2.5 Overview of data management

As already stated the data transfer will be performed by Excel^{TM} templates. Because of different needs of data, several templates form the baseline for the data transfer.

- regional data
- control strategy
- activity pathway
- other data
- emissions

Each template contains several sheets. The *main* sheet contains information which describes the whole template. This *main* sheet offers therefore the needed information to the server (e. g. user, scenario name, year, ...).

2.5.1 General

A communication is based on two partners. A receiving and transmitting is the baseline of every communication. Therefore to get information from the server, the data download, and to offer data, the data upload, is the task in which the communication is being performed. The following schema shows the user request with the corresponding action.

user $\rightarrow$ RAINS	data upload
user $\leftarrow$ RAINS	data download

Figure 2.1: Parts of communication

To get a template for uploading the user data, you will have to download an Excel™ template first.

2.5.2 Download

If the user is allowed to download data, the desired template can be figured out by parameters out of the download form. The following schema shows the flow chart of the download:

user selects the download
$\Downarrow$
checking rights
$\Downarrow$
collect the corresponding data
$\Downarrow$
fill the template
$\Downarrow$
offer download

Figure 2.2: Data download

User selects the download

With the selection of the parameters in the download form, the system can figure out the desired information out of the database. Therefore the parameters are sent to the webserver.

Checking rights

The first security check takes place. The user privileges are stored in a table in the Oracle database. Dependent on the user privileges the download will proceed. If the user is not allowed to download the required data (user has insufficient rights), an error message is stored in the database. Additional the user is informed about the missing privileges by a message displayed on the website.

Collect the corresponding data

If the user is allowed to download data from the web, the system collects the needed parameters out of the database. To increase the performance of the data collection, views are being used. A view is a collection of SQL statements which can be stored in the database. Currently some SQL statements are being stored in the JAVA sources. A modification of the desired data will cause a lot of work (change the JAVA sources, compiling the JAVA sources, restart of the web container). Therefore the SQL statements should be stored on the database side.

Fill the template

The filling of the template is one of the main tasks in the project. Currently the desired data is being filled into the template by the JAVA source code. The whole structure of the template is stored in the JAVA code. A modification of the template will therefore cause a modification of the JAVA code. It is therefore recommended to store the template structure in the database.

Offer download

The checked spreadsheet is ready for the download to the user.

2.5.3 Upload

The quality control by control strategies is the major task to ensure the quality of data. After the data has been uploaded, the stored data in the database can hardly be modified (because of the huge amount of datasets). So it is necessary to filter out as many errors and failures in the template as possible, before the upload proceeds. With the system messages (based on the control strategies) the IIASA supervisor is able to make decisions of the quality and can therefore make decisions on publishing the data to other users. The following schema shows the work flow of the data check.

user fills data into the template

$\Downarrow$

template shows up input errors

$\Downarrow$

upload of template

$\Downarrow$

supervisor check routines

$\Downarrow$

storing data in database

$\Downarrow$

reporting

Figure 2.3: Data upload

User fills data into the template

The user must own a RAINS template. Only RAINS templates are allowed for the upload to the webserver. Therefore the user will have to download such a template for a given scenario (each scenario has a baseline scenario) before he is going to upload any data. After the download, the user can fill in his own scenario into the template.

Template shows up input errors

The user can only fill in data into allowed cells. Currently no cell locking is provided, but this feature will be implemented to increase the security and bring down the risk of wrong data input. The template also provides internal checks (e. g. sum checks, if the allowed limit is exceeded, coloring the cell red).

Note: The supervisor will have to define such checks like locked cells and sum checks. That is why it is important to have a fresh and up-to-date RAINS template.

Upload of template

After the user has filled in the data into the template, the upload can proceed. In any case the user can upload the new scenario independent of any formal check (also sensitiveness tests should be allowed to be uploaded). Any occurring error (bad errors such as wrong file or format or insufficient user privileges) are displayed on the website and additionally sent to the responsible supervisor.

Supervisor check routines

The supervisor check routines are an additional main task of checking. These rules depend on the personal experiences of the responsible supervisor. These routines are being automated. Each violation and success of a supervisor rule is being stored in the database and sent to the supervisor. This will help the supervisor in making decisions of publishing the data to other users.

Storing data in database

In any way the data is being stored in the database. With the consistently storage of the data a historical data and therefore a kind of scenario birth-control is given. This stored data is always visible to the uploading user, but can only be offered to other users by the supervisor.

Reporting

A report, based on the warnings and errors of 2.5.3 are offered to the user and to the IIASA scenario-responsible supervisor. With this report the supervisor is able to make decisions on publishing the data to other users. This report can be a pdf-file, email or simply a popup window including the report.

2.5.4 Implementation

As stated the RAINS is a JAVA model, therefore all needed implementations should be done in JAVA too. To deal with spreadsheets the APACHE POI libraries (look at [ApachePOI]) could build the main base for the template handling. The POI library enables a lot of possibilities for spreadsheet modifications.

2.6 Mathematical fundamentals of the RAINS model

2.6.1 Nomenclatur

This formulas are building the baseline of emission control in the RAINS model [Alcamo et. al. 1990]. Additional formulas and directives can be found in [CAFE].

Symbol	Definition
E	Energy consumption
hv	Heat value
sc	Sulfur content
sr	Fraction of SO_2 emissions retained in ash
x	Fraction of emissions removed by pollution control
S	Sulfur emissions
S^P	Sulfur emissions for industrial (non combustion) processes
e_{NO_X}	NO_X emission factor
e_{SO_2}	SO_2 emission factor
N	NO_X emissions
N^P	NO_X emissions from industrial (non combustion) processes
sh	Shares of emissions
$scen$	Scenarios

Table 2.2: Nomenclatur for emission calculations

2.6.2 SO_2 Emission calculations (only combustion emissions)

Emission vector:

$$e_{SO_2} = \frac{sc_{i,j,k}}{hv_{i,j}}(1 - sr_{j,k})(1 - x_{i,k,l}) \tag{2.1}$$

Sectoral emissions per fuel:

$$S_{i,j,k}(t) = \sum_l E_{i,j,k,l}(t) e_{SO_2} \vee i, j, k, l \tag{2.2}$$

Total sulfur emissions per country:

$$S_i(t) = \sum_j \sum_k S_{i,j,k}(t) + S_i^P(t) \tag{2.3}$$

2.6.3 NO_X Emission calculation (only combustion emissions)

Emission factors:

$$e_{NO_X} = e(fuel - N) + e(thermal - N) + e(prompt - N) \ , \tag{2.4}$$

where $e(NO_X)$ is the total NO_X emission factor, $e(fuel - N)$ is a function of the fuel and the firing mode, $e(thermal - N)$ is a function of the combustion temperature, and stoichiometry, and $e(prompt - N)$ is negligible.

Sectoral emissions per fuel:

$$N_{i,j,k}(t) = \sum_l E_{i,j,k,l}(t) e_{NO_X} \vee i, j, k, l \tag{2.5}$$

Total NO_X emissions per country:

$$N_i(t) = \sum_j \sum_k N_{i,j,k}(t) + N_i^P(t) \tag{2.6}$$

2.6.4 Indices and their domains

Symbol	Definition
$i \in I$	Emission countries / regions
$k \in K$	Receptor countries / regions
$j \in J$	Receptor grid cells
$s \in S$	Sectors
$f \in F$	Activities
$p \in P$	Pollutants
$t \in T$	Technologies
i	Country
j	Activity type
k	Economic sector
l	Abatement technology

Table 2.3: Indices and their domains

Subsets and their representation

Symbol	Definition
$ii \in \hat{I}$	Countries
$j \in J_1$	Receptor cells in which sulphur is limiting in secondary PM formation
$j \in J_2$	Receptor cells in which nitrate is limiting in secondary PM formation
$ss \in \hat{S}$	Sectors included in the optimization
$vp \in VP \subset P$	Pollutants temporarily represented with cost curves
$p \in GHG \subset P$	Greenhouse gases
$pp \in \hat{P}$	Pollutants included in the optimization. This may be identical to P.
$vt \in VT$	Technologies represented as segments in temporary cost curves

Table 2.4: Definition of subsets

Simple subsets

The following simple subset $\vec{s}$ describes a set of subsectors applicable to the different types of powerplants: new powerplants (PP_NEW), existing powerplants(PP_EX_OTH) and wet bottom powerplants (PP_EX_OTH).

$$\vec{s} = \text{set of subsectors of } s \text{ (applicable to PP\_NEW, PP\_EX\_OTH)} \tag{2.7}$$

In particular, e. g. $\overrightarrow{\text{PP\_NEW}} = \text{ppnews}$: **subsectors of** $PP\_NEW pertaining$ **to different boiler types**

$$\text{ppnews} = \text{PP\_NEW}\alpha_{\alpha=1,2,3} \subset S. \tag{2.8}$$

the power plant sector

$$\text{POWER} = \text{PP\_NEW, PP\_EX\_OTH, PP\_EX\_WB} \subset S \tag{2.9}$$

2.6.5 Set definitions

The subset A_i in formula 2.10 describes which combinations (i, s, f) are occurring in the baseline, i. e. those which are non zero in the baseline. Moreover, in the present version of the model it also determines which combinations (i, s, f) are possible, hence it restricts the potential fuel substitutions to those that substitute occurring activities by occurring activities.

$$(s, f) \in A_i \tag{2.10}$$

Technologies controlling pollutant $p, (t, p)$

$$t \in T_p, \bigcup_p T_p = T \tag{2.11}$$

Technologies that applicable to sector/activity (s, f), controlling pollutant p

$$t \in T_{s,f,p} = T_{s,f} \bigcap T_p \tag{2.12}$$

These subsets cam also be used to derive other subsets, e. g. P_t, the set of all pollutants p controlled by technology t, or F_i, s, the activities applicable in region i and sector s.

2.6.6 Rational of data structure

In contrast to RAINS where activities stay constant and only application rates of technologies are changed in order to meet an objective in an optimization, in GAINS both activity data and application rates of technologies are variable. Since technologies may control more than one pollutant and activities may be emitting more than one pollutant it is necessary to ensure consistency across activities, technologies and pollutants. Additional complications are introduced by further disaggregation in the PM module. The consistency is achieved in the following way.

First for each activity the relevant pollutants are determined. For example, for the combination (PP_NEW, HC1) the relevant pollutants are CO_2, SO_2, NO_x, but not PM or N_2O, since these are emitted only in the subsectors PP_NEW1,2,3. Each of these pollutants may or may not be controlled by one or more technologies, each controlling a certain share of the total activity. For each pollutant p, there may be a share that is uncontrolled, and the corresponding 'technology' is labeled 'NOC_p' (in contrast to the single 'NOC' option in the database that is currently used for all pollutants), so that the sum of all shares of technologies (including the relevant no-control 'technology') is 100 %. In this way each activity $xa_{i,s,f}$ is sliced up into technology-specific activity data $x_{i,s,f,t}$, so that, if summed over all technologies that apply to a given pollutant p that is emitted from (s, f) the total activity is recovered:

$$\sum_{t \in T_p} sh_{i,s,f,t} = 1 \qquad \forall i, s, f, p \tag{2.13}$$

$$x_{i,s,f,t} = sh_{i,s,t,t} \cdot xd_{i,s,f} \qquad \forall i, s, f, t \tag{2.14}$$

$$\sum_{t \in T_p} x_{i,s,f,t} = xp_{i,s,f,p} \tag{2.15}$$

2.7 System results

End-of-pipe control cost for country i

$$\mathrm{c\_control\_cost}_i = \sum_{(ss,f)\in A_i} \sum_{t\in \overline{T}_{s,f,t}} \mathrm{unit\_control\_cost}_{i,ss,f,t} \cdot x_{i,ss,f,t} \tag{2.16}$$

Emissions of pollutant p in country i

$$\mathrm{emissions}_{i,p} = \left(\sum_{(s,f)\in A_i} \sum_{t\in T_{s,f,p}} \mathrm{EF}^{\mathrm{abated}}_{i,s,f,t,p} \cdot x_{i,s,f,t} \right) \tag{2.17}$$

2.8 Examples of possible rules

Technology potentials

If t controls any of the pollutants, then:

$$x^{min}_{i,s,f,t} \le x_{i,s,f,t} \le x^{max}_{i,s,f,t}, \qquad \forall i, \forall (s,f) \in A_i, \forall t \in T_{s,f} \tag{2.18}$$

These constraints allow to easily specify potentials of individual options as necessary in expert mode.

Emissions standards

For each sector-activity combination the average emissions per unit of activity should not increase.

$$\sum_{t\in T_{s,f,p}} \mathrm{EF}^{\mathrm{abated}}_{i,s,f,t,p} \cdot x_{i,s,f,t} \le \mathrm{IEF}^{\mathrm{BL}}_{i,s,f,p} * \sum_{t\in T_{s,f,p}} x_{i,s,f,t} \cdot 1.000001 \tag{2.19}$$

where BL[2] is the baseline scenario.

Technology standards

Per given sector-activity combination there are sub-standard technologies (e. g. EURO-II in 2020), whose share should not increase in new scenarios. In particular this applies to the respective no-control options NOC_p for each pollutant p.

$$x_{i,s,f,t} \le \frac{\mathrm{appl}^{\mathrm{BL}}_{i,s,f,t}}{100} \cdot \sum_{t'\in T_{s,f,p}} x_{i,s,f,t'}, \qquad \forall i, \forall (s,f) \in A_i, \forall t \in T_{s,f,p} \tag{2.20}$$

[2]Baseline scenarios are built out of directives, protocols and expertises

Keep number of vehicles as in the baseline

$$\sum_{f} \sum_{\text{HD\_types}} \left(\frac{x_{i,\text{HD\_type},f,\text{NOC\_CO}_2}}{\text{fuel\_per\_vehicle}_{i,\text{HD\_type},f}} \cdot 1000 - \left(\#\ \text{vehicles}_{\text{HD\_type},f}\right)_{\text{BL}} \right) = 0 \qquad (2.21)$$

Maximum share of selected alternative fuels in transport: HD and LD4

E. g. biodiesel has maximum share in total diesel (MD + BD):

$$\sum_{\text{LD4\_types}} \left(x_{i,\text{LD4\_type,MD,NOC\_CO}_2} - \frac{(1 - \text{BD\_max})}{\text{BD\_max}} \cdot x_{i,\text{LD4\_type, BD, NOC\_CO}_2} \right) = 0 \qquad (2.22)$$

Similar constraints apply to HD (biodiesel), LD4 (ethanol), LD4 (CNG), HD (hydrogen) and LD4 (hydrogen).

Chapter 3

Implementation issues

3.1 Design concept

3.1.1 Consistency checks

Two types of consistency checks (verification of data exchange) can be implemented. One of these has to rely on personal experience of the responsible manager. The more systematic approach (automated checks) can be used for verification of consistency of specific tables.

3.1.2 Automated checks

Automatic testing strategies need to allow for quick verification of internal consistency of exchanged (uploaded) datasets. For that purpose a number of specified rules has to be developed. Such rules need to be derived in collaboration with the model developers. Examples of checks could include.

- Sum of control technology percent $\leq 100\%$
- Checking for negative values where they are not expected

  ...

3.1.3 Error handling

Following the task, the system should provide transparent information (diagnostic) regarding the status of the operation. When errors are identified, database is rolled back and an error message is being displayed. It is of utmost importance that the error messages can be associated with specific problems, i. e.

- missing data

- incorrect format
- exceeding range

  ...

It is recommended that a glossary of error messages and associated causes is prepared.

3.2 Method

The first task is to develop the set of rules for the automatic checks. This will be done in consultation with the model developers. The results of discussion will be then translated into the SQL statements that will become part of the model.

Following development stage, extensive testing will be preformed to assure robustness of the rules and their implementation. In parallel user friendliness will be asserted and the output of the analysis (consistency checks) adjusted accordingly.

3.3 Technical needs

Concluding

The technical needs of the system can be summed up by the following points

- Formal consistency
- Soft consistency
- Rules definitions
- Metadata

3.3.1 Formal consistency

Formal consistency means basic logical rules in the database. These rules can only be made by database administrator (software developer). A database defined with such rules is called *relational database* [DBDesign 2001]. For database design standards you will need to decide:

- What kinds of naming conventions for database objects will you follow?
- What kinds of metadata will you capture about the database design and where will you store it?
- What procedures will you follow when designing a database from a logical data model?

- What kinds of tools will you use during the database design process (estimating tools, simulation tools)?
- What variations in scope will you allow for in your database designs (application database design versus enterprise database design)?
- Will you utilize database views?
- What procedures and naming conventions must you follow when implementing database security profiles and user IDs?

You should think about these questions before you start with a database. A very important thing is to define responsible persons for the database design. It would make no sense to have on the one hand a lot of database administrators (DBAs) and on the other hand a database name convention for each DBA. A chaos would be the result. Clearly defined rules pave the way to the goal. After the decisions of naming conventions the structure of the tables and views has to be defined. This will built the first level of data security, and in that case quality. Tables can be defined with strict defined rules.

- No value(s) allowed
- No missing value(s) allowed
- Atomic database structure

No value(s) allowed

In some cases an empty field is necessary. If the field is needed for some administrative modifications (ordernumber).

```
CREATE TABLE table_name
(column1 datatype null/not null,
 column2 datatype null/not null,
 ...
 columnN datatype null/not null,
);
```

Listing 3.1: Create table

This code will create a table *table_name*. With the parameter null or not null the administrator can decide if values are needed or not. In case of null, null values (no values) are allowed.

No missing value(s) allowed

As shown in this section the DBA has the possibility to decide if values are needed or not. In case of *no missing value(s) allowed* the table should be created in the following way:

```
CREATE TABLE table_name
(column1 datatype not null,
 column2 datatype not null,
 ...
 columnN datatype not null,
);
```

Listing 3.2: Create table no missing values allowed

JAVA allows the user to fill in data in the database. The user is connected to database by forms which are generated out of the JAVA code. If the user tries to fill in no values in values needed fields or causes any other failure, an exception is being thrown and the software developer can handle this error message.

Atomic database structure

Atomic database structure means that no redundant information is being stored in the database. The main data is being fragmented in different tables. So each table has its own unique content referenced by an index. If some content is required, it can easily being selected by this unique index.

3.3.2 Constraints

Constraints are providing another opportunity to include strict formal rules into the database. In artificial intelligence and operations research, constraint satisfaction is the process of finding a solution to a set of constraints. Such constraints express allowed values for variables, and a solution is therefore an evaluation of these variables that satisfies all constraints. [Constraints]

Definition

Constraints are one or more definitions of ranges to ensure the integrity of data in a database. They are assigned to tables and columns. Constraints are represented by the following types in Table 3.1.

Type	Description
not null	The column has to be filled. No empty field is possible.
unique	Columns or combinations of columns are the primary key. This key is unique in the table and can therefore easily and safety be referenced to another table.
primary key	A primary key is a value that can be used to identify a unique row in a table.
foreign key	A foreign key (FK) is a field or group of fields in a database record that point to a key field or group of fields forming a key of another database record in some (usually different) table.
check	This method is returns an boolean value. Depending if the value is valid.

Table 3.1: Types of constraints

3.3.3 Soft consistency

Besides strong database definitions which are being defined from the DBA, soft rules can be defined from the DBA and the supervisor. Supervisor is that person who is responsible for the correctness of data uploads. The formulas of 2.6.5 are representing such definitions for soft consistency checks.

3.3.4 Rules definitions

The rules are on the one hand depending on personal experiences of the supervisor and on the other hand already defined rules (formulas) of the RAINS model (look at 2.6).

As there is already a need of a formula interpreter for the optimization part of the RAINS model the IIASA LaTeX formula interpreter **SMT** (**S**trucuterd **M**odeling **S**ets) [SMT] could be implemented. The following Figure shows a possible interchange of RAINS and SMT.

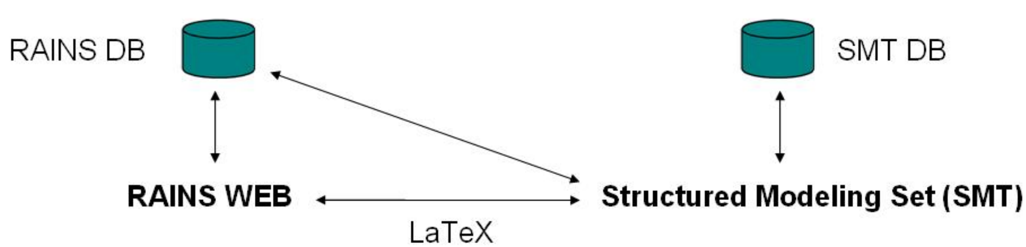

Figure 3.1: RAINS SMT interchange

3.3.5 Metadata

In parallel metadata are being stored in the database. Metadata are additional data to the upload. The database table includes the whole description of the uploaded data. The entries are being made by the supervisor who is responsible for the validation of the data.

- dataset complete?
- dataset public?
- sensitivity analysis?
- testing scenario?
- restricted used?
- valid / expiration date?

Based on the metadata the corresponding output of the uploaded scenario can be marked or signed with different alert levels. If any ”rule”, defined in the metadata, is being violated, the output can be be marked with an additional red headline (like ATTENTION data is not validated). If the user is allowed to view the data (dataset is public) an additional headline (data is valid) can be added to the output of the scenario data.

The entries are being made by the supervisor. The automated checks help the supervisor to make decisions for organization and validation of the data.

3.4 Quality judgment

As stated before the checks represent a guideline of the data quality. The uploading user is always allowed to view his data, but all the other users need to be informed about the state of the data. Therefore additional information has to be stored, which identifies the state of data. Therefore different levels of quality has to be made:

- validated
- only internal
- sensitiveness check
- test

  ...

Additionally a time stamp can be referenced to the data. So each scenario is visible to the user for a given time range. After the data has expired the scenario is only visible to the supervisor. To review the expired data a new time stamp can be added. A time stamp can offer the opportunity to publish the data to another user only for a given time range.

Chapter 4

Solutions

As stated in 2.5.4 RAINS is a Client-Server model. The model is a stable running system with JAVA as the development software and Oracle as the data storage unit. Therefore the new features should be based on these elementary premises too.

Advantage	Description
Libraries	Numerous existing libraries can easily be implemented.
Portability	Program once, Run anywhere (platform independence). JAVA runs on most major hardware and software platforms, including Windows 95 and NT, the Macintosh, and several varieties of UNIX
Security	The compiler, interpreter, and JAVA-compatible browsers all contain several levels of security measures that are designed to reduce the risk of security compromise, loss of data and program integrity, and damage to system users.

Table 4.1: JAVA advantages

In principle, each action is based on a client request and a Servlet answer. The following table shows the workflow of the data download.

User		Servlet
download request	$\Longrightarrow$	
		process
	$\Longleftarrow$	template

Table 4.2: Data download

4.1 Implementation

Spreadsheets are the basis of data exchange. The system has to fill and read the spreadsheet. Therefore the structure of the template has to be strict and known to the system. Currently the structure is hard coded in the system. That means that the structure of each template is stored in the source code. A modification of a template will cause a modification of the software, and as the source code is performed as a Servlet, a restart of the container of the webserver is needed too. A better solution would be, that the structure of the template is being stored in the database. This would cause only a data change in the database, instead of the whole procedure stated before.

The template can be modified in different ways. Most modifications are caused by an additional sub split of activities. The need of the structure in the system could be solved with a registration of the template. That means that the supervisor will have to register the new template to the system. The system can scan the new template and therefore offer the new template to the user. Then the structure must not be declared in the sources and a modification, with additional registration, will not cause any other needs of improvement.

4.1.1 Template layout

Each template contains various sheets. The name of the template (e. g. activity pathway) is the unique ID of the system. Therefore the ID of the referencing table in the database should be the name of the template. The main sheet of the template (each template owns such a ”main”sheet) implements all needed information according to the data transfer (scenario name, user name, . . .).

4.1.2 Template reading

The need to read templates is one main issue of the system (besides the writing). The Apache POI library offers some good features that can be used for the Excel™ templates. The following source code reads out a specific cell.

```
FileInputStream fis = new FileInputStream("c:/temp/test.xls");
HSSFWorkbook wb = new HSSFWorkbook(fis);
HSSFSheet sheet = wb.getSheetAt(0);
HSSFFormulaEvaluator evaluator = new HSSFFormulaEvaluator(sheet, wb);

// suppose your formula is in B3
CellReference cellReference = new CellReference("B3");
HSSFRow row = sheet.getRow(cellReference.getRow());
HSSFCell cell = row.getCell(cellReference.getCol());
HSSFFormulaEvaluator.CellValue cellValue = evaluator.evaluate(cell);

switch (cellValue.getCellType()) {
   case HSSFCell.CELL_TYPE_BOOLEAN:
      System.out.println(cellValue.getBooleanValue());
      break;
```

```
    case HSSFCell.CELL_TYPE_NUMERIC:
        System.out.println(cellValue.getNumberValue());
        break;
    case HSSFCell.CELL_TYPE_STRING:
        System.out.println(cellValue.getStringValue());
        break;
    case HSSFCell.CELL_TYPE_BLANK:
        break;
    case HSSFCell.CELL_TYPE_ERROR:
        break;
    case HSSFCell.CELL_TYPE_FORMULA:
        break;
}
```

Listing 4.1: Read value from spreadsheet

4.1.3 Template writing

To write data into the spreadsheet the Apache POI libraries offers useful methods too. The following source code writes a value into a specific cell. This value can be any type (string, float, ...).

```
// create a new file
FileOutputStream out = new FileOutputStream("workbook.xls");
// create a new workbook
HSSFWorkbook wb = new HSSFWorkbook();
// create a new sheet
HSSFSheet s = wb.createSheet();
// declare a row object reference
HSSFRow r = null;
// declare a cell object reference
HSSFCell c = null;
// create cell style
HSSFCellStyle cs = wb.createCellStyle();
HSSFDataFormat df = wb.createDataFormat();

// create font object
HSSFFont f = wb.createFont();

//set font 1 to 12 point type
f.setFontHeightInPoints((short) 12);
//make it blue
f.setColor( (short)0xc );
```

```
// make it bold
//arial is the default font
f.setBoldweight(HSSFFont.BOLDWEIGHT_BOLD);

//set cell stlye
cs.setFont(f);
//set the cell format
cs.setDataFormat(df.getFormat("#,##0.0"));

// set the sheet name in Unicode
wb.setSheetName(0, "\u0422\u0435\u0441\u0442\u043E\u0432\u0430\u044F␣
   " +
                   "\u0421\u0442\u0440\u0430\u043D\u0438\u0447\u043A\
                      u0430",
                HSSFWorkbook.ENCODING_UTF_16 );
// in case of compressed Unicode
// wb.setSheetName(0, "HSSF␣Test", HSSFWorkbook.
   ENCODING_COMPRESSED_UNICODE );
// create a sheet with 30 rows (0-29)

   // create a row
    r = s.createRow(rownum);
    c = r.createCell(cellnum);
    c.setCellStyle(cs);
    c.setCellValue( "Test" );



// demonstrate adding/naming and deleting a sheet
// create a sheet, set its title then delete it
s = wb.createSheet();
wb.setSheetName(1, "DeletedSheet");
wb.removeSheetAt(1);
//end deleted sheet

// write the workbook to the output stream
// close our file (don't␣blow␣out␣our␣file␣handles
wb.write(out);
out.close();
```

Listing 4.2: Write value to spreadsheet

4.1.4 Select data from the database

To select data from the database SQL scripts have been declared. These scripts are stored in the database. As stated in 4.1.1 the templates are different. So for each data request a query is being stored in the database.

The user request is being send to the server with some additional parameters. These parameters identifies the needed query out of all stored queries. The table `queryTables` contains all needed queries. The following code selects the necessary parameter out of the table `queryTables`.

Code for the selection of control strategy for the parameters:

- User: WebRains
- Region: Austria
- Activity type: control strategy

```
select tableQuery from queryTables where tableName='
   activityPathwayFromScenario';
```

Listing 4.3: Select the needed query out of the database

```
select * from scenario where scen=$scenarioName$
   AND owner=$scenarioOwner$
   AND region=$regionAbb$
   AND act_type=$typeList$
```

Listing 4.4: The result

The $variables$ ($scenarioName$, $scenarioOwner$, ...) of the Listening 4.4 are variables. This variables are being replaced with the parameters of the user request.

Now with the given parameters the system can select the needed parameters and values out of database. The queries for the selection are also stored in the table `queryTables`. The following sources selects the needed data out of the database to fill the required spreadsheet.

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.sec_abb like 'TRA_RD%'
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
```

Listing 4.5: Select parameters for road transport sources

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.sec_abb like 'TRA_OT%'
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.6: Select parameters for off road vehicles and machines

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.sec_abb like 'DOM%'
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
```

Listing 4.7: Select parameters for coal combustion

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.sec_abb like 'DOM%'
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.8: Select parameters for biomass combustion

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.emived_SO2 = 1
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.9: Select parameters for SO_2

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
```

```
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.noxemved = 1
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.10: Select parameters for NO_X

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and ( a.sec_abb like 'AGR%' or a.sec_abb='FERTPRO' or a.sec_abb='
    FCON_UREA' )
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.11: Select parameters for NH_3

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.pmemived = 1
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.12: Select parameters for PM

```
SELECT c.act_abb, c.sec_abb, c.tech_abb, c.year, c.perc FROM constr_n
    c, act_sec_tech a
 WHERE con_strat='cl_aust_whol_Apr04'
 and a.VOC_eved = 1
 and c.act_abb=a.act_abb
 and c.sec_abb=a.sec_abb
 and c.tech_abb=a.tech_abb
ORDER BY c.act_abb, c.sec_abb, c.tech_abb
```

Listing 4.13: Select parameters for VOC

4.1.5 Storing data in database

To store data into the database queries are also needed. The selection of the query is similar to the section stated before in 4.1.4. The user uploaded data (spreadsheet) is scanned. That means that JAVA routines are reading the values of the spreadsheet. After successfully reading the data is stored in the database. As shown in the following table, JAVA builds the bridge from the user data (spreadsheet) to the server (database).

spreadsheet	$\Longleftarrow$ JAVA $\Longrightarrow$	database

Table 4.3: Workflow of data storage

The tables `act_sec_tec` and `constr_n` are containing all data for control strategy. Therefore new scenario data must be stored in these tables. To store a value in the tables the following SQL 4.14 script has to be performed.

```
INSERT INTO constr_n(CON_START,ACT_ABB,SEC_ABB,TECH_ABB,YEAR,PERC)
values("name","abbreviation","sector","type","year","value")
```

Listing 4.14: Store value for control strategy

Name	Description
Name	is the name of the new controlstrategy (declared in the template as upload_name)
Abbreviation	the type of the data (e. g. BC1 (brown coal / lignite, gread 1))
Sector	the occurring cause (e. g. CON_COMB (fuel production))
Type	types of occurring emission (referenced to table act_sec_tec)
Year	the year of the uploading value
Value	the value (percentage)

Table 4.4: Description of parameters

4.2 Database

As there is already a running system, the checking system has to be integrated into the existing database structure. Instead of rebuilding already defined and existing tables (e. g. usertables), such tables can be implemented and be used for the system. The following Fig. 4.1 is a small extract of the database structure presenting the user tables.

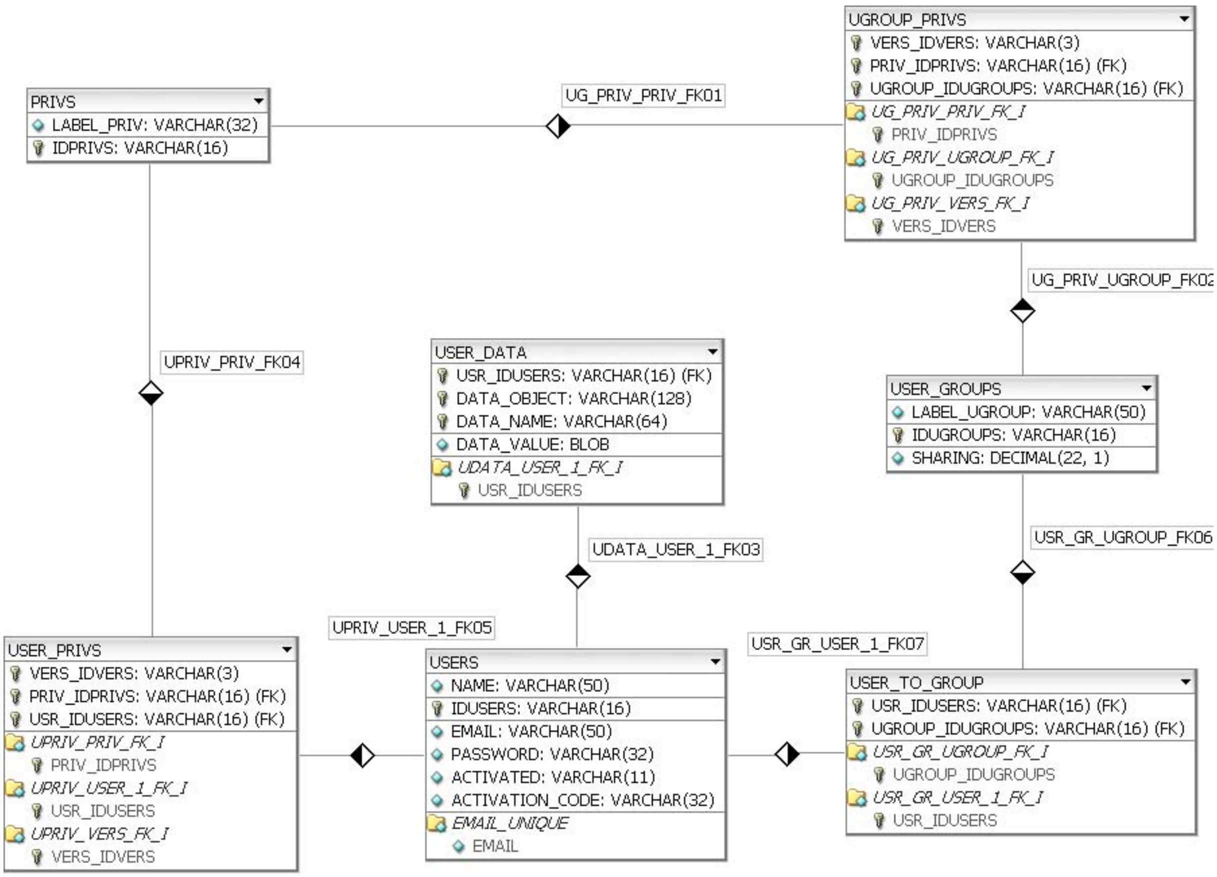

Figure 4.1: Overview user tables

The table USERS can adapted to the new system. This table contains all necessary information of the currently registered users. The other tables (i. e. USER_GROUPS) are used in the current system for handling the users and their privileges. The metadata are being stored in the database. Yet no specific tables had been implemented. To append the needed structures (Figure 4.2) the following code examples are being performed:

```
CREATE TABLE tabMetaReports (
  idtabMetaReports INTEGER UNSIGNED NOT NULL,
  idtabMetadata INTEGER UNSIGNED NOT NULL,
  Report BLOB NULL,
  PRIMARY KEY(idtabMetaReports, idtabMetadata)
);
```

Listing 4.15: Create the table tabMetaReports

```
CREATE TABLE tabMetadata (
  idtabMetadata INTEGER UNSIGNED NOT NULL,
  DATA_NAME VARCHAR(64) NOT NULL,
  DATA_OBJECT VARCHAR(128) NOT NULL,
  USR_IDUSERS VARCHAR(16) NOT NULL,
  userID INTEGER UNSIGNED NOT NULL,
  userAddress VARCHAR(20) NULL,
  filename VARCHAR NULL,
  idReport INTEGER UNSIGNED NULL,
  type_2 INTEGER UNSIGNED NULL,
  PRIMARY KEY(idtabMetadata),
  INDEX tabMetadata_index1371(userID)
);
```

Listing 4.16: Create the table tabMetadata

Now the new structure (Fig. 4.2) is available.

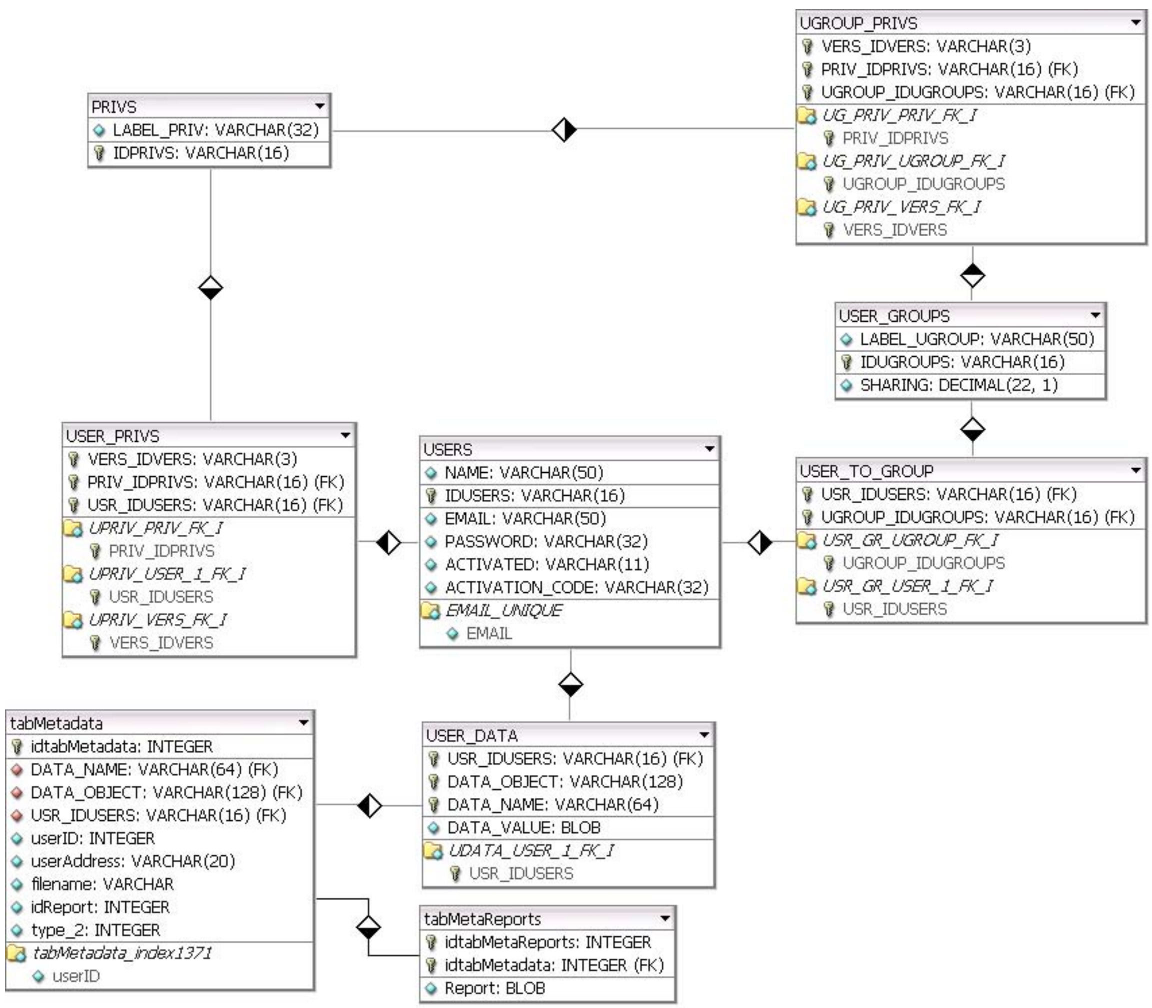

Figure 4.2: New structure of Usertables

4.3 Checks

4.3.1 General

The system checks can be split into 2 parts (Fig. 4.3). On the one hand the strict rules depend on mathematical and statistic knowledge (e. g. sums $\leq$ 100 %) and on the other hand the soft rules, which depend on personal experiences of the supervisor.

$$
\begin{array}{ccc}
 & \text{rules} & \\
 & \swarrow \quad \searrow & \\
\text{strict rules} & & \text{soft rules} \\
\text{(only database)} & & \text{(database and-or template)}
\end{array}
$$

Figure 4.3: Rules overview

The storage location can either be the database or any other type of storage (such a XML files). The first generation of the system checks will use the database as backup for the rules. The next generations can be used with extensions to XML files.

4.3.2 Database

To store the supervisor rules, the new tables have to be created and implemented into the system. The first task is to define all possible checks and create therefore a table structure.

Definition of tables structure

To define the structure the basic test has to be declared. The easiest way of checking is to check if the value is null (and allowed to be null).

Check on null

To reference a field out of a spreadsheet 2 coordinates are needed. The corresponding structure contains therefore 4 values. The x- and y- coordinate, the name of the spreadsheet (remember there are different types) and a boolean field which shows up if null values are allowed. Because of the huge amount of data and the knowledge that there are only a few values where null values are not allowed. The corresponding table only contains such values where no nulls are allowed. Therefore only 3 fields are needed. The following script generates such a table in the system.

```
CREATE TABLE chkNoNullAllowed
(spreadsheetName VARCHAR(40) not null,
 xCoor VARCHAR(20) not null,
 yCoor VARCHAR(20) not null
);
```

Listing 4.17: Create the table chkNoNullAllowed

Range check single value

To check the ranges can offer a new possibility of checking. A value can either be within limits or out of the limits. As stated before 2 values are needed to reference the value. To enter a range two additional values are needed: the upper and the lower limit. Therefore 5 fields are needed: The x- and y- coordinates, the name of the spreadsheet, the upper and the lower limit. The following source code will create the needed table structure.

```
CREATE TABLE chkRange
(spreadsheetName VARCHAR(40) not null,
 xCoor VARCHAR(20) not null,
 yCoor VARCHAR(20) not null,
 lowerLimit VARCHAR(20) not null,
 upperLimit VARCHAR(20) not null
);
```

Listing 4.18: Create the table chkRange

Range check sum of values

To validate a sum of the values it is recommended to build the sum in the template. Therefore only the, already summed up value, needs to be read. Instead of reading each value, summing it up and finally check the sum in the system.

Trend

The more exclusive way of checking is to analyse the trend of the data. The values of uploading emission data should be lower than already stored data of emission in further years. Therefore a trend analysis is possible. There is no need of extra tables in the database. The trend can be built out of already existing data tables only the result of the analysis should be stored in the database.

Improvement

To reduce the number of tables and therefore save storage on the database server. The 2 checking tables can be added to 1 resulting checking table. The following script will perform the new table ”tabChecking”.

```
CREATE TABLE tabChecking
( spreadsheetName VARCHAR(40) not null,
   xCoor VARCHAR(20) not null,
   yCoor VARCHAR(20) not null,
   lowerLimit VARCHAR(20) null,
   upperLimit VARCHAR(20) null,
   nullAllowed VARCHAR(20) null
);
```

Listing 4.19: Create the table tabChecking

Name	Description
spreadsheetName	is the name of the uploading spreadsheet
xCoor	is the x coordinate of the to be checked value
yCoor	is the y coordinate of the to be checked value
lowerLimit	is the lower allowed limit of the value
upperLimit	is the upper allowed limit of the value
nullAllowed	references of a null value is allowed (YES)

Table 4.5: Description of parameters

Chapter 5

Examples

The following example explains the generation (filling) of the template with a possible checking routine.

5.1 Explanations

To get a better overview of the template the spreadsheet is colored (look at Fig. 5.1). Only blank fields (white colored) are allowed to be modified. The color codes have the following meaning:

Combinations with gray background are not allowed in RAINS

Values are calculated automatically (using data from the same worksheet).

Values are calculated automatically (using data from other worksheet).

Energy - total			Upload:	no	**Units:**	**PJ**									
Upload nam	bleu43_Apr	Owner	IIASA	Region	AUST_WHOL										
year	Act_abb	CON_COI	CON_LOS	IN_BO	IN_OCTO	DOM	TRA_RD	TRA_OT	TRA_OTS	PP_EX_W	PP_EX_O	PP_NEW	PP_TOTA	NONEN	SUM
1990	BC1	0	0	3.4	0.4	6.3	0	0.1	0	0	23	0	23	0	33.2
1990	BC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HC1	0	7	0.2	7	4.1	0	0.3	0	0	39.6	0	39.6	0	58.2
1990	HC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HC3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	DC	0	0	0	34	16.2	0	0.4	0	0	0	0	0	0	50.6
1990	OS1	0	0	22.4	5.3	102.7	0	0	0	0	4.6	0	4.6	0	135
1990	OS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HF	2	4	8.1	10.2	33.9	0	0	0	0	26.1	0	26.1	0	84.3
1990	MD	0	0	0.1	0.2	53.1	71.02	18	0	0	0.9	0	0.9	0	143.32
1990	ETH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	GSL	0	0	0	0	0.129	108.46	14.27	0	0	0.6	0	0.6	0	123.459
1990	LPG	0	0	0	1.9	4.07	0.5	0	0	0	0	0	0	0	6.47
1990	MTH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	GAS	37	3	33.1	28.3	53.5	0	0	0	0	98.2	0	98.2	0	253.1
1990	H2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	REN	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HYD	0	0	0	0	0	0	0	0	0	0	0	112.17	0	112.17
1990	NUC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	ELE	0	21	0	65	80	0	9.5	0	0	0	0	-178	0	-2.5
1990	HT	0	3	0	3	20	0	1.9	0	0	0	0	-27.9	0	0
1990	**Sum**	**39**	**38**	**67.3**	**155.3**	**373.999**	**179.98**	**44.47**	**0**	**0**	**193**	**0**	**99.27**	**0**	**997.319**

Figure 5.1: Screenshot activity pathway of energy and mobile

5.1.1 Template for activity pathway

This spreadsheet contains data on activities in energy and transport sector. The spreadsheet includes the following sheets:

- Main - contains technical parameters necessary to upload information to RAINS WEB - ignore when providing data
- En_tot - national energy use by fuel and sector
- En_mob - fuel use by mobile sources
- Veh_km - annual mileage by road transport sources
- Veh_no - number of vehicles
- Air_dom_sh - share of aviation fuels consumed by domestic air traffic (part included in the national assessment of CO_2 emissions)
- Biof_tr_sh - share of biofuels in total fuel use by mobile sources
- En_ren - details on consumption of renewable fuels
- En_proc - activity of processes linked to energy pathway (currently cement and lime production)
- En_pr_param - parameters for energy-related processes

5.2 Data request

To download the activity pathway for energy and transport the user can enter the needed parameters on the HTML page (Fig. 5.2) of the RAINS WEB.

With the following parameters the user selects the needed options to download the necessary data:

Option	Parameter
Part	Activity pathway for energy and mobile
Version	Aug04
Scenario	BL_CLE_Apr04
Region	AUSTRIA

Table 5.1: Download parameters

The needed queries are stored in the table `queryTable`. With the parameters of the download selection (Table 5.1) the needed queries are selected out of the database. The following example query selects the needed values for `En_tot` out of the database.

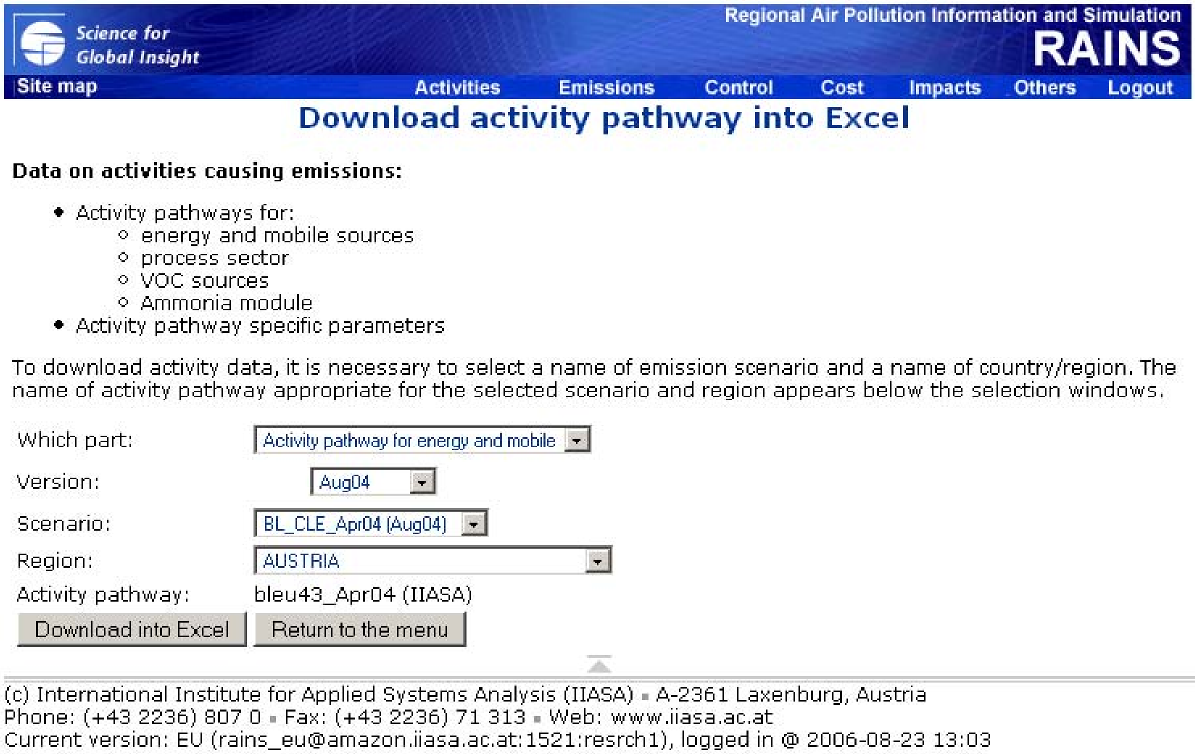

Figure 5.2: Download activity pathway for energy and mobile

```
select y.year, e.act_abb,
 sum(e.CON_COMB*nvl(a.activity,0)) as CON_COMB,
 sum(e.CON_LOSS*nvl(a.activity,0)) as CON_LOSS,
 sum(e.IN_BO*nvl(a.activity,0)) as IN_BO,
 sum(e.IN_OCTOT*nvl(a.activity,0)) as IN_OCTOT,
 sum(e.DOM*nvl(a.activity,0)) as DOM,
 sum(e.TRA_RD*nvl(a.activity,0)) as TRA_RD,
 sum(e.TRA_OT*nvl(a.activity,0)) as TRA_OT,
 sum(e.TRA_OTS*nvl(a.activity,0)) as TRA_OTS,
 sum(e.PP_EX_WB*nvl(a.activity,0)) as PP_EX_WB,
 sum(e.PP_EX_OTH*nvl(a.activity,0)) as PP_EX_OTH,
 sum(e.PP_NEW*nvl(a.activity,0)) as PP_NEW,
 sum(e.PP_TOTALB*nvl(a.activity,0)) as PP_TOTAL,
 sum(e.NONEN*nvl(a.activity,0)) as NONEN, e.sequ
 from excel_en_tot e join year y on y.year=y.year
 left outer join actpath a
  on a.act_abb=e.act_abb and  a.sec_abb=e.sec_abb and
     a.year=y.year and
    a.path_abb='bleu43_Apr04' and
    a.owner='IIASA' and
```

```
    a.region='AUST_WHOL'
group by e.act_abb, y.year, e.sequ
order by y.year, e.sequ
```

Listing 5.1: Select values

The result of the SQL query is represented in the following figure.

#	YEAR	ACT_ABB	CON_COMB	CON_LOSS	IN_BO	IN_OCTOT	DOM	TRA_RD	TRA_OT	TRA_OTS	PP_EX_WB	PP_EX_OTH	PP_NEW	PP_TOTAL	NONEN	SEQU
1	1990	BC1	0	0	3.4	.4	6.3	0	.1	0	0	23	0	0	0	1
2	1990	BC2	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	1990	HC1	0	7	.2	7	4.1	0	.3	0	0	39.6	0	0	0	3
4	1990	HC2	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5	1990	HC3	0	0	0	0	0	0	0	0	0	0	0	0	0	5
6	1990	DC	0	0	0	34	16.2	0	.4	0	0	0	0	0	0	6
7	1990	OS1	0	0	22.4	5.3	102.7	0	0	0	0	4.6	0	0	0	7
8	1990	OS2	0	0	0	0	0	0	0	0	0	0	0	0	0	8
9	1990	HF	2	4	8.1	10.2	33.9	0	0	0	0	26.1	0	0	0	9
10	1990	MD	0	0	.1	.2	53.1	0	0	0	0	.9	0	0	0	10
11	1990	ETH	0	0	0	0	0	0	0	0	0	0	0	0	0	11
12	1990	GSL	0	0	0	0	.129	0	0	0	0	.6	0	0	0	12
13	1990	LPG	0	0	0	1.9	4.07	0	0	0	0	0	0	0	0	13
14	1990	MTH	0	0	0	0	0	0	0	0	0	0	0	0	0	14

Figure 5.3: Query result

5.3 Filling the template

All needed parameters have been selected (Fig. 5.3) and are ready to be filled into the template. Each template has a given structure (e. g. activity pathway):

Energy - total			Upload:	no	Units:	**PJ**									
Upload nam	bleu43_Apr	Owner	IIASA	Region	AUST_WHOL										
year	Act_abb	CON_COI	CON_LOS	IN_BO	IN_OCTO	DOM	TRA_RD	TRA_OT	TRA_OTS	PP_EX_W	PP_EX_O	PP_NEW	PP_TOTA	NONEN	SUM
1990	BC1	0	0	3.4	0.4	6.3	0	0.1	0	0	23	0	23	0	33.2
1990	BC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HC1	0	7	0.2	7	4.1	0	0.3	0	0	39.6	0	39.6	0	58.2
1990	HC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HC3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	DC	0	0	0	34	16.2	0	0.4	0	0	0	0	0	0	50.6
1990	OS1	0	0	22.4	5.3	102.7	0	0	0	0	4.6	0	4.6	0	135
1990	OS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HF	2	4	8.1	10.2	33.9	0	0	0	0	26.1	0	26.1	0	84.3
1990	MD	0	0	0.1	0.2	53.1	71.02	18	0	0	0.9	0	0.9	0	143.32
1990	ETH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	GSL	0	0	0	0	0.129	108.46	14.27	0	0	0.6	0	0.6	0	123.459
1990	LPG	0	0	0	1.9	4.07	0.5	0	0	0	0	0	0	0	6.47
1990	MTH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	GAS	37	3	33.1	28.3	53.5	0	0	0	0	98.2	0	98.2	0	253.1
1990	H2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	REN	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	HYD	0	0	0	0	0	0	0	0	0	0	0	112.17	0	112.17
1990	NUC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1990	ELE	0	21	0	65	80	0	9.5	0	0	0	0	-178	0	-2.5
1990	HT	0	3	0	3	20	0	1.9	0	0	0	0	-27.9	0	0
1990	**Sum**	**39**	**38**	**67.3**	**155.3**	**373.999**	**179.98**	**44.47**	**0**	**0**	**193**	**0**	**99.27**	**0**	**997.319**

Figure 5.4: Template structure

As shown in Fig. 5.3 the results are in a similar structure. Therefore the data can easily be filled in by using the references given by the description. For example the value for *brown coal/lignite, grade 1* in the year *1990* for the *domestic* sector can be referenced by even these parameters out of the query result.

5.4 Checks

The implementation of the checks (stated in 3.3) can be implemented as shown in the following examples.

5.4.1 Soft checks

The softchecks can be placed in the spreadsheets or in the database. The speadsheet based checks can be implemented by the supervisor. The database based soft checks can only be implemented by the supervisor with additional backup of a DBA.

Spreadsheet based soft check

The soft checks stated in 3.3.3 can be implemented in the template itselfs. The supervisor can enter visual checks with conditional formating [SoftCheck]. This kind of check allows the supervisor to check sums and single values of their magnitude, value and range. The following checks ,concerning the cell value, are possible:

- between / not between
- equal / not equal to
- greater / less than
- greater than or equal to
- less than or equal to

With conditional formating cells can be colored and can be used to highlight violated rules. The spreadsheet-based soft checks can be implemented by the supervisor and the developer. After the modification of the template (adding of the Conditional Formating) the template has to be stored on the webserver (replacing the old template).

Database based soft checks

These checks are also simple checks on the parameter‘s magnitude and value. The rules defined in the database are checking the uploaded template and generate an error report. With such an error report the supervisor will not have to look on the spreadsheet for checking.

The activity pathway template represents an energy balance of the corresponding scenario. Each power plant (coal-burning, solar-power, . . .) produces electricity by fuel-transformation.

An uploaded scenario can be checked with an existing baseline scenario. To produce the needed electricity fuels are needed. Each country produces energy with the regional occurring fuels (e.g in Poland is coal the main power source). Therefore the sum of the fuels of all powerplants (look at 2.9) is defined as

$$\sum_{s \in P} \sum_{f \in F} xd_{i,s,f} \,. \tag{5.1}$$

The sum of the produced electricity is defined as

$$\sum_{s \in P} xd_{i,s,f,ELE} \,. \tag{5.2}$$

The production of electricity depends on the efficiency factor of the used fuels [EUROGAS]. For the same amount of energy supplied ,natural gas generates less CO_2 than other fossil fuels.

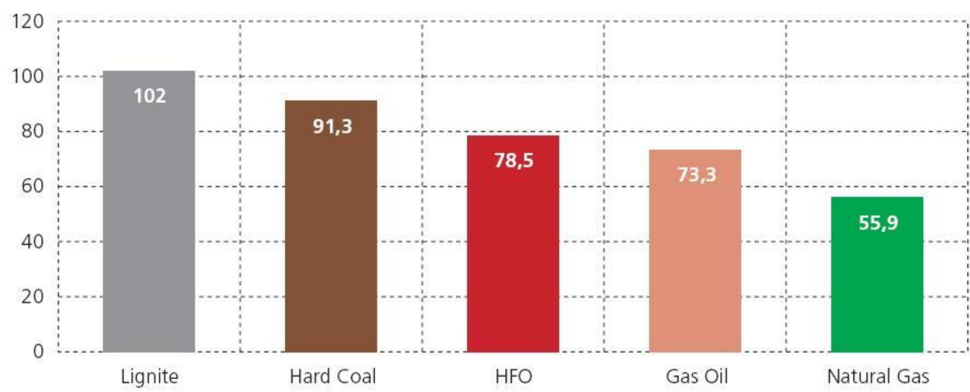

Figure 5.5: CO_2 formed by the combustion of fossil fuels (kg CO_2 / GJ)

The efficiency factor of the fuels is very important for the electricity production. A fuel change may cause a reduction or an increase of electricity production and the corresponding reduction or increase of emissions.

For example 100 PJ coal generates 30 PJ electricity. That states an efficiency of $\frac{100}{30}$ $100\,\%$ = $30\,\%$. 60PJ of gas are producing 30 PJ of electricity. That is an efficiency of $50\,\%$.

A change of coal to gas increase the energy production of the powerplant and reduce the emissions. Therefore a reduction of emissions can be done by using another,more efficient, fuel. The difference from the sum of the used fuels (Formula 5.2) to the sum of the fuels of the baseline scenario can be checked with

$$diff_{FUEL}[\%] = \frac{\sum_{s \in P} \sum_{f \in F} xd^{BL}_{i,s,f} - \sum_{s \in P} \sum_{f \in F} xd^{scen}_{i,s,f}}{\sum_{s \in P} \sum_{f \in F} xd^{BL}_{i,s,f}} \tag{5.3}$$

A modification of the fuels and the corresponding change of the efficiency causes a change of the production of the energy

$$diff_{ELE}[\%] = \frac{\sum_{sinP} xd^{BL}_{i,s,ELE} - \sum_{sinP} xd_{i,s,ELE}}{\sum_{sinP} xd^{BL}_{i,s,ELE}} \tag{5.4}$$

The formula (5.3, 5.4) results shows the grade of modification and can therefore offer the possibility of checking on the deviation to the baseline scenario.

5.4.2 Strict checks

Strict checks have to rely on mathematical and statistical rules. These rules are only stored in the database and can therefore only be modified by a system developer. The following strict rule is based on the activity pathway template.

$$0 \leq \sum E_{t,i} \leq \sum E$$

Where E is the energy of a scenario and a specific technology per year. This can‘t be negative or more than 100 % (the sum of the energy of all technologies). Another strict rule could be

$$\sum E_t \leq \sum E,$$

where E_t is the sum of energy of the special technology and this sum must be lower than the whole energy consumption of all technologies.

Chapter 6

Conclusion

The RAINS model is a emission and cost control simulation model. To have the possibility to simulate different scenarios a lot of data are necessary. To get this enormous amount of data, either the user or IIASA stuff have to upload the data into the system. Because of the extension to the RAINS model with the Asian version (RAINS-Asia) the amount of data is growing very fast. To handle and validate this huge amount of automated data checks are being implemented. These checks helps the IIASA supervisor to validate the scenarios and the corresponding data and to make decisions on publishing the data to other users. These checks can either be implemented in the templates, which builds the baseline of the data transfer, or in the database. The result of the checks is being presented to the user and the supervisor in form of a pdf file, an email or simply by a popup window on the website of the RAINSWEB model. The implementation based on the given structure of the database offers the opportunity to deal with already implemented tools (e. g. login key, existing tables, ...).

6.1 Outlook

The implementation of formulas dependent on personal experiences with XML as a possibility which can be used for a more structured system. Also other methods like the implementation and interpretation of LaTeX formulas could be a good option. SMT could build a good baseline for such an implementation.

As stated is the modification of templates a problem in the system. Therefore the template registration should be implemented to offer on the one hand the supervisor the possibility of easy template modification with additional on time availability of new published templates and on the other hand to improve the system usability.

With these control strategies a baseline is given. On this baseline other strategies can be built. The strategies in this paper will be implemented and fully tested by the end of 2006. The actual version of the RAINS model is available on *www.iiasa.ac.at*.

Glossary

API An application programming interface (API) is the interface of a computer system, library or application provides to allow requests for services by other computer programs and / or to allow data to be exchanged between them.

$C/C++$ The programming language C (often, just C) is a general-purpose, procedural, imperative computer programming language developed in the early 1970s by Dennis Ritchie for use on the Unix operating system. It has since spread to many other operating systems, and is now one of the most widely used programming languages. C also has had a great influence on many other popular languages, especially C++ which was originally designed as an enhancement to C. It is distinguished for the efficiency of the code it produces, and is the most commonly used programming language for writing system software, though it is also widely used for writing applications. Though not originally designed as a language for teaching, and despite its somewhat unforgiving character, C is commonly used in computer science education, in part because the language is so pervasive.

CMS A content management system is a computer software system for organizing and facilitating collaborative creation of documents and other content.

$Client$ A client is a computer system that accesses a (remote) service on another computer by some kind of network. The term was first applied to devices that were not capable of running their own standalone programs, but could interact with remote computers via a network. These dumb terminals were clients of the time-sharing mainframe computer.

DBA The database administrator (DBA) is a person who is responsible for the environmental aspects of a database. Although not strictly part of a database administrator's duties, logical and physical design of databases is sometimes part of the job. These functions are traditionally thought of as being the duties of a database analyst or database designer.

$DBMS$ database management system (DBMS) is a system or software designed to manage a database, and run operations on the data requested by numerous clients. Typical examples of DBMS use include accounting, human resources and customer support systems. DBMSs have more recently emerged as a fairly standard part of any company back office.

$Database$ A database is a shared collection of logically related data designed to meet the information needs of an organization. The term originated within the computer industry, but its meaning has been broadened by popular use, to the extent that the European Database Directive (which creates intellectual property rights for databases) includes non-electronic databases within its definition. This article is confined to a more technical use of the term; though even amongst computing professionals, some attach a much wider meaning to the word than others.

ID Identifiers (IDs) are lexical tokens that name entities. The concept is analogous to that of a name. Identifiers are used extensively in virtually all information processing systems. Naming entities makes it possible to refer to them, which is essential for any kind of processing.

$JAVA$ JAVA is an object-oriented programming language developed by James Gosling and colleagues at Sun Microsystems in the early 1990s. Unlike conventional languages which are generally designed to be compiled to native code, JAVA is compiled to a bytecode which is then run (generally using JIT compilation) by a JAVA virtual machine.The language itself borrows much syntax from C and C++ but has a simpler object model and fewer low-level facilities. JAVA is only distantly related to JavaScript, though they have similar names and share a C-like syntax.

$MathML$ Mathematical Markup Language (MathML) is an application of XML for representing mathematical symbols and formulas, aimed at integrating them into World Wide Web documents. It is a recommendation of the W3C math working group. The specification of version 1.01 of the format was released in July 1999 and version 2.0 appeared in February 2001. In October 2003, the second edition of MathML Version 2.0 was published as the final release by the W3C math working group.

$Oracle$ An Oracle database, strictly speaking, consists of a collection of data managed by an Oracle database management system or DBMS. The term *Oracle database* sometimes refers - imprecisely - to the DBMS software itself. The title of this article - and parts of the article content - perpetuate this confusion. One can refer to the Oracle database management system unambiguously as Oracle DBMS or (since it manages databases which have relational characteristics) as Oracle RDBMS.

PDF Portable Document Format (PDF) is a file format proprietary to Adobe Systems for representing two-dimensional documents in a device independent and resolution independent fixed-layout document format. Each PDF file encapsulates a complete description of a 2D document (and, with the advent of Acrobat 3D, embedded 3D documents) that includes the text, fonts, images, and 2D vector graphics that compose the document. PDF files do not encode information that is specific to the application software, hardware, or operating system used to create or view the document. This feature ensures that a valid PDF will render exactly the same regardless of its origin or destination (but depending on font availability).

SQl SQL (commonly expanded to **S**tructured **Q**uery **L**anguage) is the most popular computer language used to create, modify, retrieve and manipulate data from relational database management systems. The language has evolved beyond its original purpose to support object-relational database management systems. It is an ANSI/ISO standard.

Server In information technology, a server is a computer system that provides services to other computing systems-called clients-over a network. The term is most commonly applied to a complete computer system today, but it is also used occasionally to refer only to the hardware or software portions of such a system.

Servlets The JAVA Servlet API allows a software developer to add dynamic content to a Web server using the JAVA platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the JAVA counterpart to dynamic web content technologies such as CGI, PHP or ASP. Servlets can maintain state across many server transactions by using HTTP cookies, session variables or URL rewriting. A Servlet is an object that receives requests (ServletRequest) and generates a response (ServletResponse) based on the request. The API package javax.servlet.http defines HTTP subclasses of the generic servlet (HttpServlet) request (HttpServletRequest) and response (HttpServletResponse) as well as an (HttpSession) that tracks multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a Web application.

XML The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. In other words, XML is a way of describing data. An XML file can contain the data too, as in a database. It is a simplified subset of Standard Generalized Markup Language (SGML). Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. Languages based on XML (for example, Geography Markup Language (GML), RDF/XML, RSS, Atom, MathML, XHTML, SVG, EAD, Klip and MusicXML) are defined in a formal way, allowing programs to modify and validate documents in these languages without prior knowledge of their particular form.

List of Tables

List of Figures

.

Bibliography

ADB *Asian Development Bank*. `http://www.adb.org`. – date: 18.08.06

Alcamo et. al. 1990 ALCAMO, J. ; SHAW, R. ; HORDIJK, L.: *The RAINS Model of Acidification, Science and Strategies in Europe.* Kluwer Academic Publishers, Dordrecht, The Netherlands., 1990

Amann et. al. 2004 AMANN, M. ; COFALA, J. ; KLIMONT, Z. ; SCHÖPP, W.: *Modelling of emissions.* IIASA, 2004. – Review 2004

ApachePOI *JAVA libraries*. `http://jakarta.apache.org/poi/`. – date: 18.08.06

Barret and Sandnes 1996 BARRET, K. ; SANDNES, H.: *Transboundary Acidifying Air Pollution calculated transport and exchange across Europe, 1985-1995 Transboundary Air Pollution in Europe.* Meteorological Sythesizing Centre - West, Norwegian Meteorological Institute, Oslo, Norway., 1996. – MSC-W Status Report 1996

Bertok et. al. 1993 BERTOK, I. ; COFALA, J. ; KLIMONT, Z. ; SCHÖPP, W. ; AMANN, M.: *Structure of the RAINS 7.0 Energy and Emissions Database.* IIASA, 1993. – WP-93-67

CAFE *Clear Air For Europe*. `http://ec.europa.eu/environment/air/cafe/index.htm`. – date: 18.08.06

Constraints *Definition*. `http://en.wikipedia.org/wiki/Constraint_satisfaction`. – date: 18.08.06

DBDesign 2001 INC. knowledge partners: *design of relational databases.* The Business Rules Management Company, 2001. – `http://www.kpiusa.com/BRBook/Files/DesigningRelationalDatabases.pdf`. – 9/42

EBRD *European Bank for Reconstruction and Development*. `http://www.ebrd.org`. – date: 18.08.06

EEA(EuropeanEnvironmentalAgency) 1996 EEA(EUROPEANENVIRONMENTALAGENCY): *Joint EMEP/CORINAIR Atmospheric Emission Inventory Guidebook.* EEA, 1996. – First Edition

EUROGAS *Eurogas Statistics 2003*. `http://www.eurogas.org`. – page 8,date: 18.08.06

IIASAOptions *Internal Report*. http://www.iiasa.ac.at/Admin/INF/OPT/Summer98/description.htm. – date: 18.08.06

Kyoto *Kyoto Protocol*. unfccc. – http://unfccc.int/kyoto_mechanisms/items/1673.php. – date: 18.08.06

LRTAP 2002 *Strategies and policies for air pollution abatement*. UN, 2002. – http://www.unece.org/env/lrtap. – 2002 Review prepared under The Convention of Long-range Transboundary Air Pollution

MacDonald 1998 MACDONALD, G. J.: *Controlling Transboundary Air Pollution*. IIASA, 1998. – http://www.iiasa.ac.at/Admin/INF/OPT/Summer98/feature.htm. – IIASA Options Magazine

Maran 2003 MARAN, R.: *Teach Yourself Visually Microsoft Excel 2003*. Visual, 2003. – ISBN 0-76453-996-5

Moussiopoulus et. al. 2004 MOUSSIOPOULUS, N. ; HELMIS, C.G. ; FLOCAS, H.A. ; LOUKA, P. ; ASSIMAKOPOULUS, C.Nares ; SAHM, P.: *A modelling method for estimating transboundry air pollution in southeastern of Europe*. Environmental Modeling and Software 19/6, 2004. – 549

ORACLEXE *oracle express edition*. http://www.oralce.com/. – date: 18.08.06

Posch et. al. 1997 POSCH, M. ; HETTELINGH, a J.-P. ; SMET, P.A.M. de ; DOWNING, R.J.: *Calculation and mapping of critical thresholds in Europe*. Coordination Center for Effects, RIVM, Bilthoven, The Netherlands., 1997. – Status Report

RAINS Options 1998 *Cleaner air for a cleaner future: Controlling transboundary air pollution*. IIASA, 1998. – Options

SIMDEF *Simulation definition*. http://en.wikipedia.org/wiki/Simulation. – date: 18.08.06

Simpson 1993 SIMPSON, D.: *Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989.Model results and comparison with observations*. Atmospheric Environment, 1993. – vol. 27A,p.921-943

SMT *SMT (structured modeling sets)*. http://www.iiasa.ac.at/~marek. – date: 18.08.06

SoftCheck *Conditional Formatting*. http://blogs.msdn.com/excel/archive/2006/02/24/538875.aspx. – date: 18.08.06

Wagner et. al. 2005 WAGNER, F. ; AMANN, M. ; BERGLUND ; COFALA, J. ; HÖGLUND-ISAKSOON, L. ; HEYES, C. ; KLAASSEN, G. ; MECHLER, R. ; TOHKA, A. ; SCHÖPP, W. ; WINIWARTER, W.: *The extension of the RAINS Model to Greenhouse Gases*. IIASA, 2005. – IR-04-014, International Institute for Applied System Analyses (IIASA), Laxenburg, Austria

WORLDBANK *Worldbank*. http://www.worldbank.org. – date: 18.08.06