NEEDS ASSESSMENT AND CONCEPTUAL DESIGN OF THE NILE BASIN DECISION SUPPORT SYSTEM CONSULTANCY

NILE BASIN DECISION SUPPORT SYSTEM

FINAL REQUIREMENT ANALYSIS AND DSS DESIGN REPORT

ANNEX E: Nile Information System

Vienna/Austria, Addis Ababa/Ethiopia, 17 March 2008

NILE BASIN INITIATIVE
Initiative du Bassin du Nil

hydrOphil
consulting & knowledge development GmbH

ITAD

CEDARE

NEEDS ASSESSMENT AND CONCEPTUAL DESIGN OF THE NILE BASIN DECISION SUPPORT SYSTEM CONSULTANCY

By Andreas Blumauer and Andreas Koller

With contributions from
Reinold Seidelmann

Consortium lead by
hydrophil - consulting & knowledge development GmbH
A-1160 Vienna, Austria, Lerchenfelder Gürtel 43, Top 6.3, **OFFICE:** +43 (0)1 49 464 59-0, **FAX:** +43 (0)1 49 464 59-40
www.hydrophil.at

Table of Content

1 Introduction / Project Description 1
1.1 Goals and Objectives 1
1.2 Scope of the Nile-IS 1
1.3 Key Functions (General Functional User Requirements) 1
1.4 Assumptions and Dependencies 2
1.5 Criteria for Success 2
2 Interface Structure 4
2.1 Interface Structure Front-End 4
2.2 Interface Structure Back-End 10
3 Information Access 11
3.1 System Users, User Groups and Access Rights 11
3.2 Login / Registration Process 12
3.3 Full Text Search 13
3.4 Alert System – Email Notification 13
4 Data Model and Description 14
4.1 Data Objects (Asset Types) 14
4.2 Data Dictionary and Relationships 14
4.3 Metadata Standards and Feedback Mechanism 14
5 System Specification 16
5.1 System Architecture 16
5.2 Content Management System conX 17
5.3 System Requirements conX 2.5 17
5.4 System Requirements Search Engine 17
6 Other Requirements 19
6.1 Performance Requirements 19
6.2 Security Requirements 19
6.3 Screen Design 19
7 Future Directions, Hosting and Maintenance 20

Appendix D1 – Technical Documentation conX (based on Software Framework conX 2.5)

Appendix D2 – Asset Type Specification

Appendix D3 – Process of Development

Versions	Submitted
Draft Annex E	10 January 2008
Draft Final Annex E	29 February 2008
Final Annex E	17 March 2008

Acronyms

CMS	Content Management System
CPU	Central Processing Unit
LAMP	Linux, Apache, MySQL, PHP
Nile-IS	Nile Information System
PHP	Hypertext Preprocessor
RAM	Random Access Memory
RSS	Really Simple Syndication
SQL	Structured Query Language
WAI	Web Accessibility Initiative
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 Introduction / Project Description

1.1 Goals and Objectives

The Goals of the Nile-Information System

The Nile Information System (Nile-IS) is a framework of the Knowledge Base (KB) with essential features and sufficient flexibility for scaling up to the full-fledged Nile Basin Knowledge Base. The following will specifically be supported.

- Making all information available to NBI, the identified stakeholders, and the interested public (world) in a transparent and well structured manner on a central platform. This implies that information obtained will be available for subsequent tasks and projects both within and outside NBI. This will assure effective utilization of accumulated information and increase efficiency (Included information currently only comprises information obtained and generated during this project).
- Making Users of the Nile-IS aware of the implications of the information in the system, especially with respect to applicability an quality
- Data exchange with other systems (like the website www.nilebasin.org)
- Different Access levels
- Workflows that guarantee a high level of information quality

Non Goals of Nile-IS

- A fully fledged Document Management System (with Lock-Mechanisms on any documents to be edited; containing all the documents generated throughout the whole communication processes of the NBI)
- Communication-Tools like Groupware systems usually provide

1.2 Scope of the Nile-IS

Nile-IS provides an easy-to-use platform which enables all participants of NBI to store, retrieve and exchange information in a structured way.

High quality of data will be achieved by feedback mechanisms and reliability by implementing a sophisticated concept which supports information flow from inner circles of the community out to the public.

Through the implementation of state of the art technologies, a scalable and future-proof system will be developed.

1.3 Key Functions (General Functional User Requirements)

The general layout / key functions of Nile-IS are as follows:

- Content archiving (structured document repository with search functionalities) and versioning
- A metadata database
- A tool for the storage and exchange of information between NBI units and stakeholders
- A web portal for general public (linked from NBI website)

- The web based back-end offers an easy-to-learn and user-friendly (Microsoft Windows like user interface) administration environment where

  - Metadata and assets can be administrated
  - Documents can be uploaded
  - Users and user groups and their access rights can be administrated and
  - Log Files of the system can be seen

- All content of Nile-IS is organised as assets. Any asset is a subject to a defined asset type. To any asset, one or more assets of the type "Document" can be linked. Documents can contain either unstructured (e.g. Text formatted as Adobe PDF, MS Word etc.; Images formatted as JPEG, GIF etc.) or semi-structured data (e.g. Datasets formatted as MS Excel etc.) and will always be described by a special asset type called "Document" (therefore extensive use of Dublin Core Metadata is made).
- All metadata collected with respect to any asset are stored in the conX DB-System and are indexed together with the documents (except certain document types like pictures) in the conX Index-System (which is based on Lucene).
- At the web based front-end (which is optimised for IE 5.0+ and Firefox 2.0+) any asset for which the user has access rights can be browsed and searched by using a high performance and scalable search engine, useable in two ways

  - Simple Google-like Search (by typing in a search phrase)
  - Moderated Search based on basic metadata like selecting countries, sub basins, themes or types of information (asset types)

1.4 Assumptions and Dependencies

Nile-IS was developed by re-using well and regularly deployed software components. All components are open source but have been customized especially for this project.

1.5 Criteria for Success

The client has attached great importance to the scalability of the system. Future extension of the system, both in the scope of functionalities and in the data volume, will not be restricted by design constraints but are possible in a modular manner.

- Conceptual design of an (intelligent) web-based information system
- Screen design (style guide) for efficiency information systems (including usability & accessibility)
- Technical implementation of the web-based information system
- High level Web-hosting/-housing
- Know-how on conception and implementation of intelligent search engines
- Know-how on conception and implementation of semantic web components

Further criteria

- Consulting along the whole project schedule
- Conceptual design of the system (structure of content and information, system architecture, performance, robustness, scalability, security, screen design usability, accessibility)
- Precise search mechanism to achieve high quality output of information
- Development of a target-oriented (screen) design according to community and the audience

Self-evident – but important criteria

- User friendly interfaces – consistent guidance for the user
- Media-compatible preparation of content
- High performance of the web-based system
- Definition of window-headers and meta-tags to be indexed by other search engines
- Testing (e.g. operating systems, different browser)
- Tight project management and frequently & transparent communication and
- co-ordination of the project
- WAI conformity (as needed) by using XHTML technology.

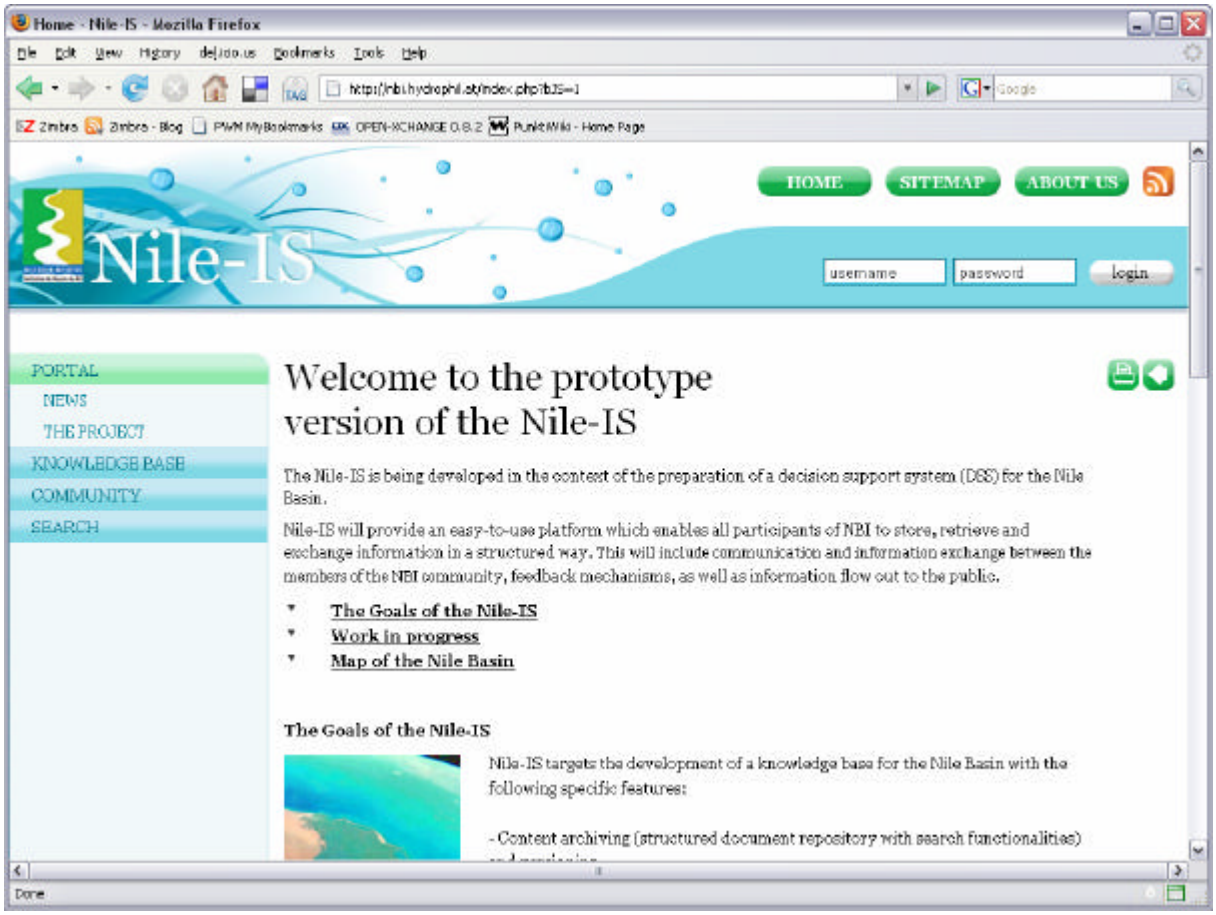
2 Interface Structure

2.1 Interface Structure Front-End

The 4 basic elements of the Nile-IS front end are:

- Portal View
- Community / User Groups
- Search
- Additional menu containing basic information about the Nile-IS

Figure 2.1 Portal View



The menu structure of the portal view can be organised comfortably in the backend where folders (and their names) define the structure in the public view (front-end). Only a two level hierarchy under the portal node is planned. In any “Topic Folder” must be one asset of the “basic information” type which will be displayed in the public view (front-end) when chosen. To publish documents in these folders, a symbolic link (“hard-links”) can be created which points to the original asset (the original asset is a part of the community). Hence, any changes on the original information will be seen also in the public portal view.

Any asset which can be displayed for a user can also be printed out.

Any page in the Nile-IS can be recommended by email. The system will generate an email which contains the link to the recommended page in the Nile-IS and send it to the entered email address.
The output at the front end will be programmed with a strong orientation towards the WAI guidelines of the W3C, thus a text only version will be offered for the most important features of the system.

Additional menu containing basic information about Nile-IS
At any stage of the front end the additional menu can be accessed:

- Home ? brings the user to the entry page (home page) of the website “basic information”
- About Us ? “basic information”
- Sitemap ? shows all available navigation links in the portal and search view – navigation links in the community space will not be shown in the sitemap (e.g. profile, community, groups, …)
- RSS ? RSS-Feed (using RSS 1.0) of the last 100 assets which were published for the public view can be accessed

“Bread crumb” navigation (pathway) is available to advance the usability of the system by making navigation more transparent:

E.g.: HOME > KNOWLEDGE BASE > INFRASTRUCTURE

The User is actually at the Infrastructure main page. By clicking on HOME, the user jumps to the main page (Home Page), by clicking on KNOWLEDGE BASE the user jumps to the Knowledge Base main page etc.

Community View
Once logged into, the Community area will offer new features (when anonymous users click on “Community”, a text which explains the advantages of an account and how to apply for an account is displayed; this text has to be determined):

- My Profile:

The user can change any information he or she has inserted when the account was created, especially the password.

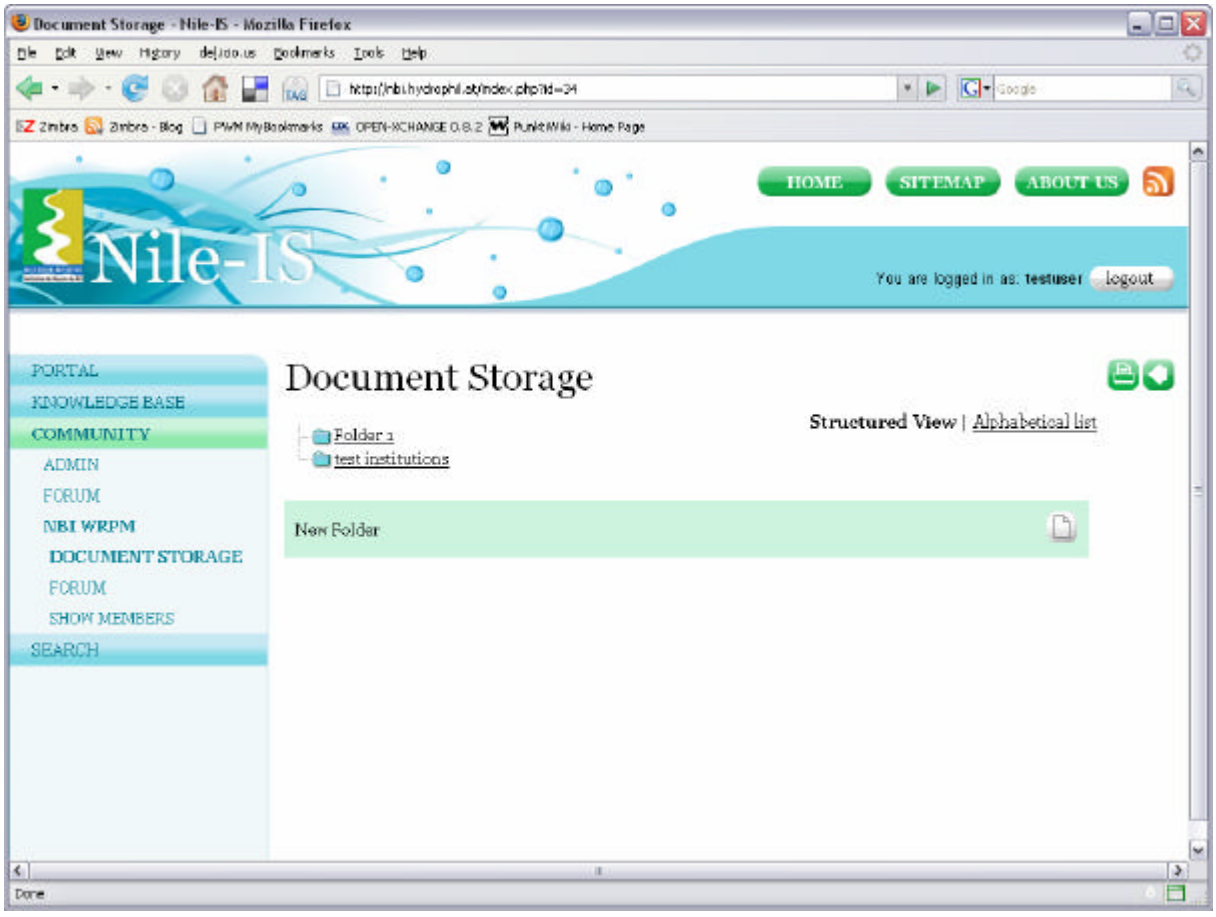
- Community:

Knowledge Base: to browse the knowledge base a Windows explorer-like interface will be offered. The user can create new assets, therefore an empty template will be displayed which conforms to the structure of the chosen asset type. Clicking on existing assets, shows well structured metadata and any linked documents can be explored and downloaded (exception: the user does not have the right to read the linked documents but the rights for the “parent-asset”. In this case the title of the document will be displayed. However when the title is clicked on, it will produce a message which informs about the restriction).
Show Members: A list of all members of the community will be shown. The members list contains first name and surname and can be filtered on initials of the members’ surname (e.g. ALL A B C D E F …). A click on the member’s name will lead to the detailed description of the member’s profile (Asset Type User).
Forum: An overview of topics will be shown and a new topic can be added (“add new topic”). Users can add postings on topics as shown in the figure below. Users may enter a subject and a message for each topic or posting.

Figure 2.2 Structure of the Forum

User Groups: shows a list of all user groups with their short descriptions and a button to apply for the membership in a certain user group (see section “Workflows”).

Figure 2.3 Community View



Adding Assets

A) Add new asset by a user group member to a user group's knowledge base

Basic principle: any member of a user group is allowed to add assets to this user group.

1. User selects his user group
2. User selects the menu tab "DOCUMENT STORAGE" inside of this user group
3. User selects the needed folder where the new asset should be added
4. User selects option: "Create Asset"
5. User selects an asset type the user wants to add out of existing available asset types (e.g. institution, spatial data, document etc)
6. User fills out the given form which is defined for the asset type (title, lead, summary, country etc) and uploads a document (if needed)
7. User saves the new asset
8. User is the owner of this new asset (means that the user has all rights on this asset)
9. New asset is available for the whole user group (read only)

The workflow of adding a new asset to the Community is exactly the same as the above described to a user group.

B) Publish an asset from user group's knowledge base to a higher level: community view knowledge base

Basic principle: administrators of user groups have all rights within this user group. Several administrators can be assigned to a user group (by back end super users).

- User group administrator selects asset to publish (out of the user groups knowledge base – out of the relevant folder)
- User group administrator selects function "publish asset" (only available for user group administrators)
- User group administrator fills out a form "publish request" and sends this request to the community administration (e.g. why this asset needs to be published for the community)
- Community administration receives an internal message about the new asset publishing request.
- Community administration has 2 choices:

    a. Accept publishing request → Community administration selects "accept request" → An internal message is sent by the system automatically to the requesting user group administrator containing a publishing confirmation → Community administration has to set a hard link inside of the back-end (CMS) onto the relevant asset to publish it inside of the Community. → The asset is now available in the Community's knowledge base (for all members of the Community – read only)

    b. Deny publishing request → Community administration selects "deny request" → An internal message is sent by the system automatically to the requesting user group administrator containing a publishing denial (this message is editable by the Community management – e. g. to give a reason of the publishing denial)

The workflow of publishing an asset from the Community to the public view is exactly the same as the above described from the user group to the Community.

Join User groups

To become a new member of a user group it is necessary to be logged in. The next step is then to apply for membership (part of the submenu “List of user groups”) for a selected user group. All administrators of this user group receive an email with a short message (to be defined) which contains the name and email of the user who applied for membership. The administrator can now log in to the system and add the user to the user group which generates an email with a short notification (to be defined) of the new membership. This mail will be sent to all administrators of the user group and to the new member

Search

Users (no matter if logged in or anonymous) can search those assets (and therefore also those documents) they have the access rights for. There are two ways to search the knowledge base:

- Simple search: Like Google a text field will be offered. The user can type in one or more words (separated by blanks). When the search button is clicked the search engine will look for assets (entries) where the search phrase can be found. If more than one word is typed in, all the words must be found in an entry to be displayed. Quotation marks around two or more words mean that the exact phrase must be in the asset to be shown in the search results. The search results will be displayed as a list.
- Moderated search: In addition to the free search phrase the user can set filters: By selecting countries, sub basins, themes and asset types, the search results can be narrowed down and hence the time to find suitable results can be reduced.

Figure 2.4 Search View

The query interface makes also multiple selections of attributes possible: e.g. a query can be based on more than one country, sub basin or theme or a combination of all of them.

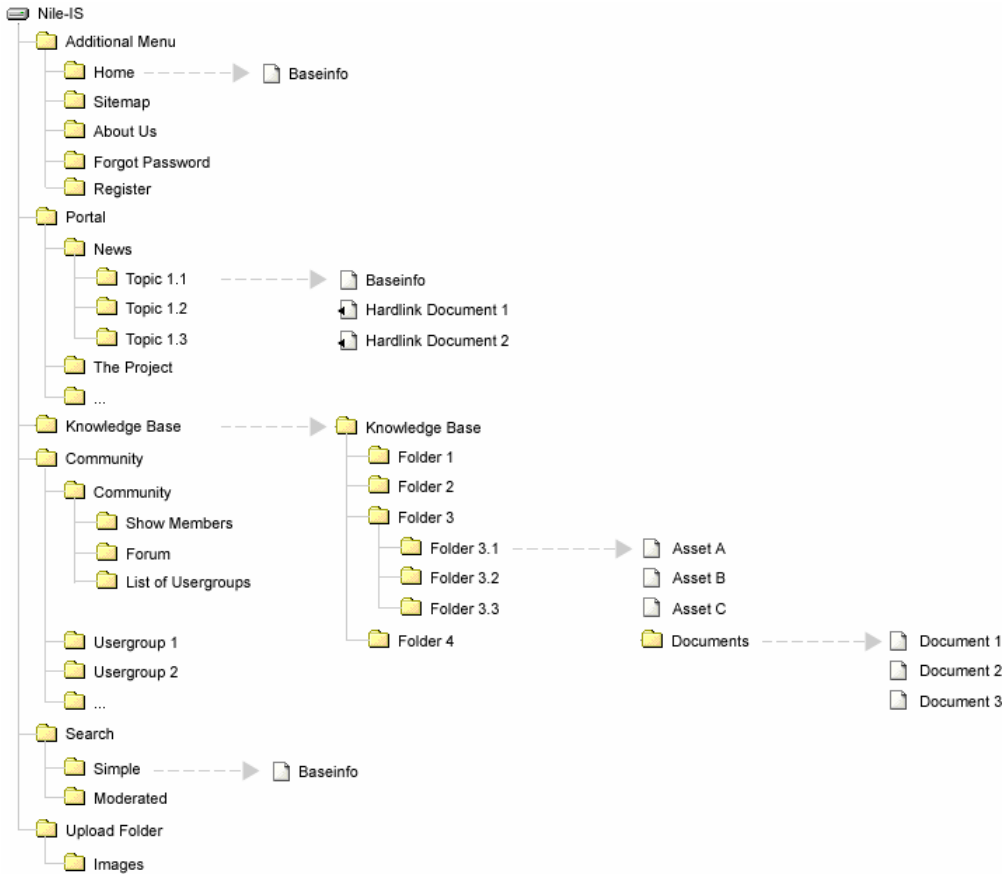
Moderated search: selecting a country or a sub basin

A country and/or a sub basin can be selected by clicking on the country / sub basin map. The selected country / sub basin will be highlighted. Only one country / sub basin can be chosen.

Figure 2.5 Nile Countries / Sub Basin Map

Interface Structure Back-End
Folders (and their names) define the structure in the public view (front-end). In any folder there must be an asset type “Basic information” which will be shown if chosen in the front-end. The content of “Home” e.g. must be entered in the asset type “basic information” in the folder “Home”, information about the Sitemap must be entered in the asset type “basic information” in the folder “Sitemap” etc.

Figure 2.6 Overview of the Backend Folder Tree

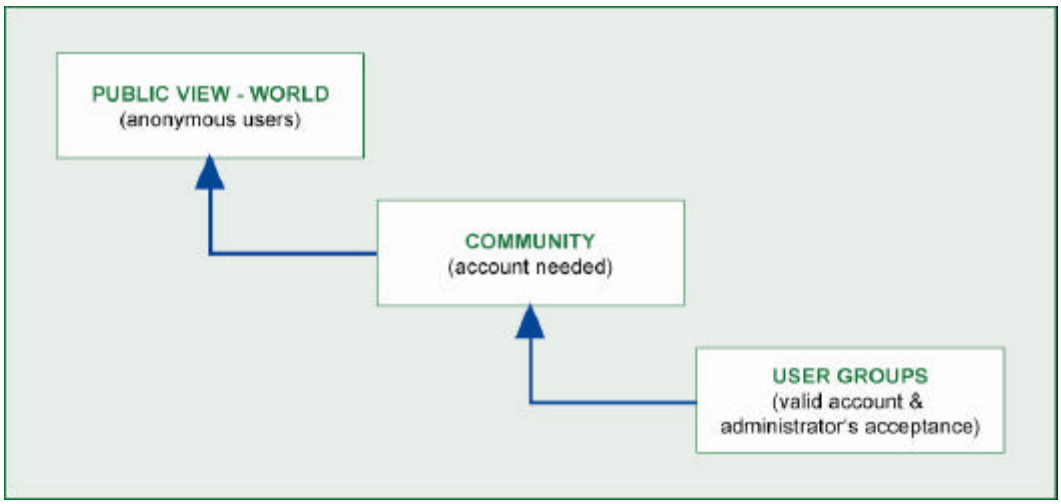


Allowed asset types in the folders and subfolders in the knowledge base are: Institutions, Infrastructure, Project, Treaty, Time Series Data, Database, Spatial Data and Model.

3 Information Access

3.1 System Users, User Groups and Access Rights

At the front end there are 3 levels of access:



For the back-end, super users can be defined who have unrestricted access to all information. They also can grant any access rights to any other user if they want.

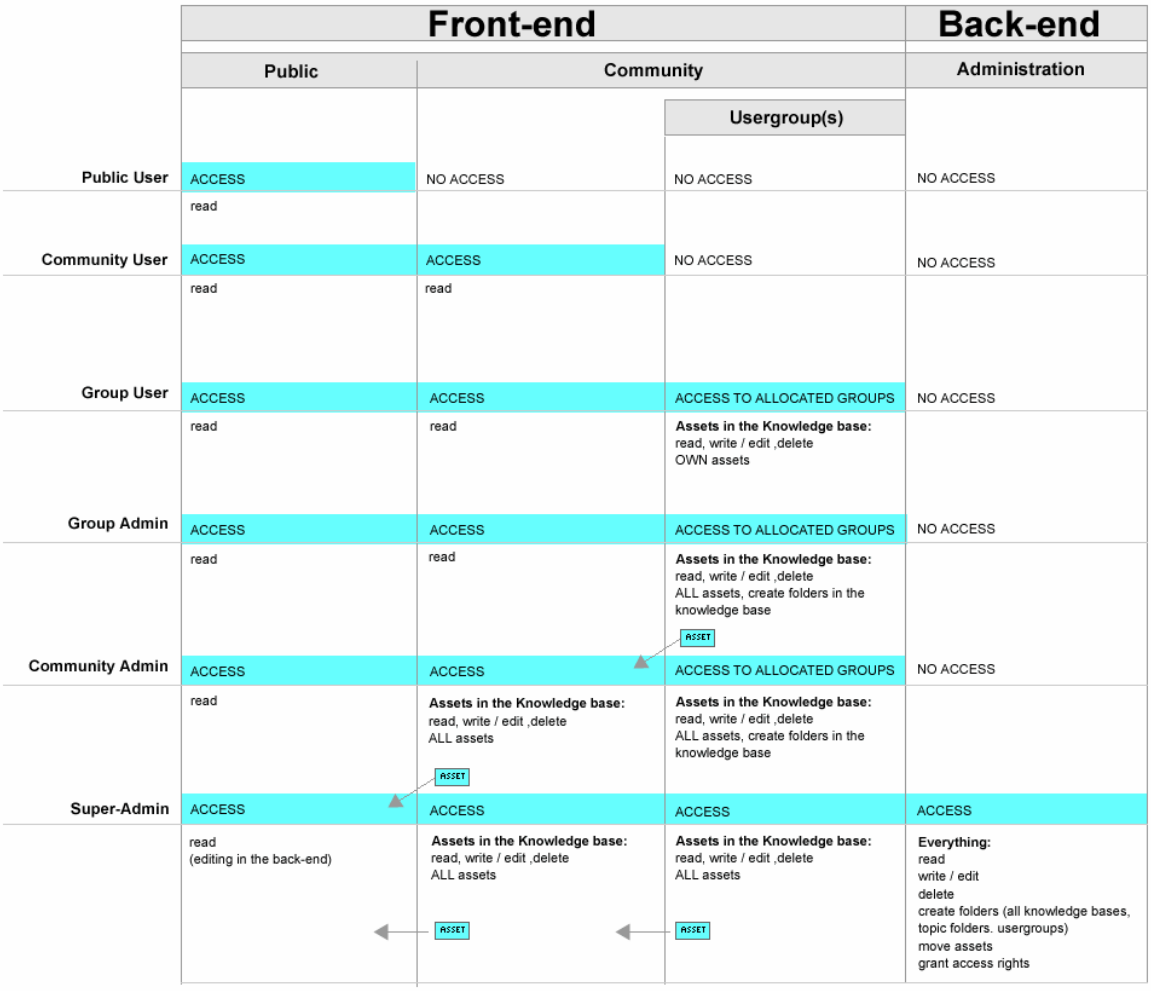
For any user group which only can be defined by super users, administrators (at least one or more) can be defined. Besides unrestricting rights on all assets inside “his or her” user group, the administrator can decide if several assets should be readable also for the community level (the the community level is designed following the same principle). The creator of an asset has any right on his or her asset except lifting it up to the next level.

Within the Nile-IS back-end the following access rights can be applied to any folder (assets inside the folder inherit the defined rights) and/or asset to any user group (and therefore to their members) or single users by the back end super users:

- See existence of asset (but no right to read or open)
- Read
- Write / Edit
- Delete
- Rename
- Move (and therefore put assets to a higher level and publish them)
- Grant access rights

The administration of the above mentioned access levels are handled via the Nile-IS back-end super users.

Figure 3.1 Overview of the Access Rights



3.2 Login / Registration Process

Login

To login for the community the user clicks on “login” and must fill in a form with the following fields:

- Email
- Password

The number of attempts to login successfully is not limited. After the successful login the user will be redirected to the community view / navigation point “Community”.
If the user *forgot his password* he can click on “Forgot your password” and a new automatically generated password for his account will be generated and sent to his email account.

Registration

Users wishing to enter community space have to register at the Nile-IS.
To *create an account* for the community level (NOT for the user groups!) the user clicks on “Create new account” and must fill in a form with the following mandatory fields:

- First name
- Last name
- E-Mail

In addition optional information can be inserted:

- Institution
- Language skill
- Job title

After the form was sent, the user receives an email with a welcome message, an automatically generated password, the username (which is the email-address) and a link. By clicking on this link, the account will be activated and the user can log in.

3.3 Full Text Search

Boolean operators and Wild cards

- The "AND" operator is the default conjunction operator. The "AND" operator matches documents where all terms exist anywhere in the text of a single document
- The "OR" operator links two terms and finds a matching document if either of the terms exists in the text of a single document
- The "NOT" operator excludes documents containing the term after "NOT"

  The "NOT" operator cannot be used with just one term (e.g. searching for “NOT Egypt” will return no results).
- The “*” symbol to perform a multiple character wildcard search

Phrase search
A phrase is a group of words surrounded by double quotes (e.g. “Nile water”). The use of wild cards and Boolean operators is not possible on phrase searches.

Search result
The search result list contains:

- Asset Title
- Text passages containing the search terms
- Metadata (not completely defined yet)

The search result page contains 25 entries. If more entries are found a page turning function will be shown (e.g. “< 1 2 3 >”)

3.4 Alert System – Email Notification

The Nile-IS alert system sends every week an email to registered users with links to new added assets Only users with respective access rights are notified. Every registered user has the option to activate or deactivate the alert system in his own user profile.

4 Data Model and Description

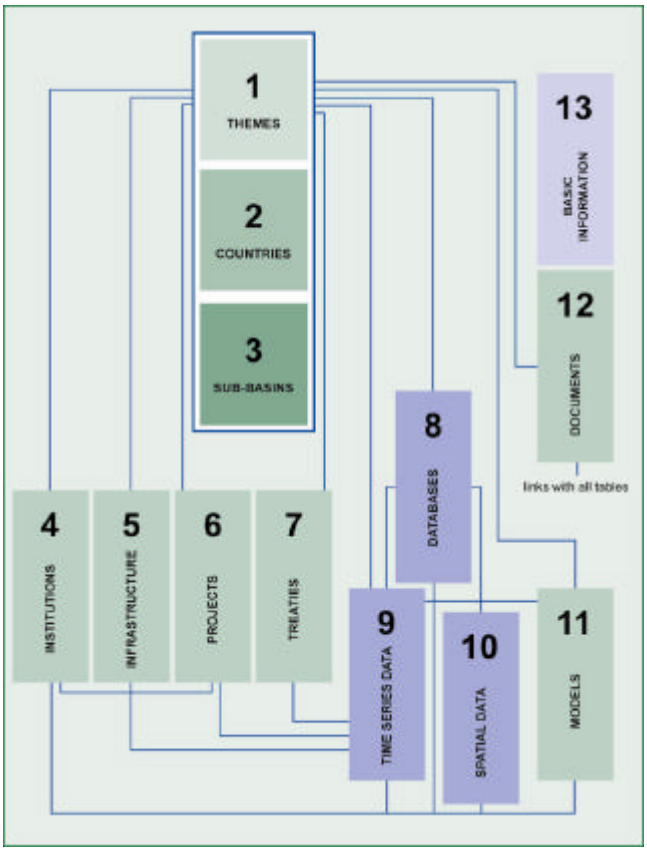
4.1 Data Objects (Asset Types)

Please refer to Appendix D2 “Asset Types Specification” for exact Asset Types definition.

4.2 Data Dictionary and Relationships

The underlying data model is defined by so called “asset types”, their attributes and their m:n-relations to each other.

Figure 4.1 Data Model



4.3 Metadata Standards and Feedback Mechanism

Due to the wish of the client to implement a quality assurance system, three special metadata attributes should be noted here:

- Any document can be rated by users who are logged in. By simply clicking on “thumb up/thumb down”1 they can express if a document should be considered of high or low

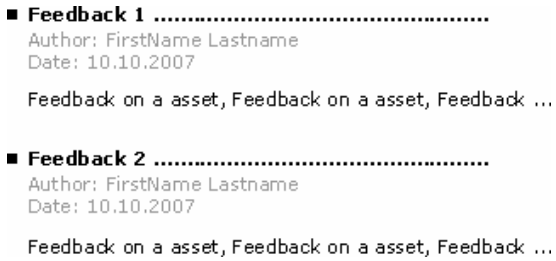
[1]See https://slynkr.sunwarp.net

quality. An average value (which is between 1 and 5 stars) will be calculated. First Example: After a document was added to the system (starting with 3 “stars”), three people clicked “thumb up” (which equals 5 “stars”), one person clicked “thumb down” (which equals 1 star), and maybe 10 others did not click at all. This will give a result of (1*1+3*5+1*3)/5 = 3,8 which will be rounded up to 4 “stars”. Second example: 6 persons voted “bad”, 11 voted “good” which results in (6*1+11*5+1*3)/18 = 3,56 (again rounded up to 4 “stars”). Every user can only rate an asset once.

Figure 4.2 Vote Button

- Any registered user can attach feedback (free text) to any asset he or she wants. Feedback will be displayed at the front end for any user who has access to the asset. Users can add feedback on assets as shown in the figure below. User may enter a subject and a message.

Figure 4.3 Feedback on an Asset



- For any asset of the system a full history of all versions will be stored and can be used for recovery

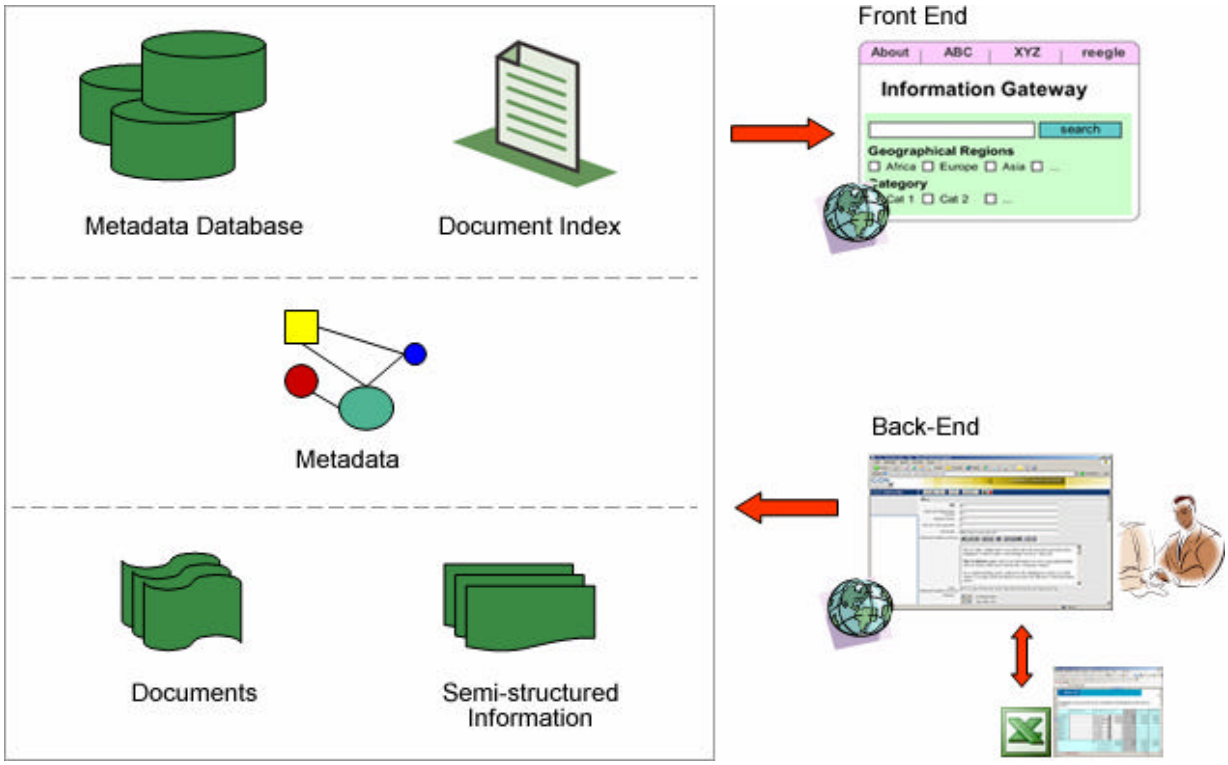
5 System Specification

Both the client's system environment and the used system conX 2.5 are based on LAMP: Linux operating system with Apache webserver, MySQL database and PHP programming language. The system to be set up is scalable and flexible enough to accommodate large amounts of data.

Data exchange with other systems / databases can be managed through XML interfaces. In a first step an RSS feed will be implemented.

5.1 System Architecture

Figure 5.1 System Architecture



5.2 Content Management System conX

The system is implemented on top of the Content & Community Management System called "conX". conX (which is implemented for the widespread LAMP-architecture) has been developed since 2001 and is available in version 2.5. It has been applied on several international projects having similar purposes to this (Reegle.info, EUWI.net, EUWI-CIS, Ecoi.net, PWM.at, Semantic-Web.at, …).

For more information please see Appendix D1.

5.3 System Requirements conX 2.5

The current CMS conX 2.5 is using a LAMP architecture:

- Linux
- Apache web server
- MySQL database
- PHP as coding language

Web Server
Apache web server >= 2.0

- PHP5 -> 5.2.0
- PHP5-gd
- PHP5-MySQL
- PHP5-XML
- Gettext 0.10.40-5

Database

- MySQL -> 4.0.24

Further needed

- ASPELL for spellchecking

Hardware

- CPU >= 1 GHz
- RAM >= 1GB

5.4 System Requirements Search Engine

System requirements (search servlet)

- Java SE Runtime Environment
- Application server (jetty-5.1.12)
- Database server (mysql)
- Lucene-2.2.0
- Mounted file directory

- cron

The full text search engine will index following content:

- All assets in the database
- pdf (not encrypted, not password protected)
- doc (Microsoft word documents)
- html (Hypertext documents)
- txt (Text documents (.txt, .rtf))

Languages:

- English
- French
- or any language using latin character sets

Encoding:
Any standard-encoding like ISO-8859-1, UTF-8, Windows-1252

Document size:
< 20MB

Architecture
Main asset types (4,5,6,7,13, …) will be indexed along with their country, topic and sub-basin information (1,2,3) into single lucene index-documents. Access rights as well as asset type information will also be saved in those. Index-documents resulting in the following index-document structure for a single instance of an asset type:

- Lucene Document
  - (Lucene field) Countries
  - Countries
  - (Lucene field) Themes
  - Themes
  - (Lucene field) Subbasins
  - Subbasins
  - (Lucene field) Assettype
  - One of (Institutions, Infrastructure,...)
  - (Lucene field) Content
  - Content (all fields) of asset type along with all attached documents and files
  - (Lucene field) Access rights
  -

making it possible to have the following full text search possibilities:

- General Full text search - content of all asset types (all countries, all topics, all sub-basins) will be searched
- General Full text search with constraints content of the given
  - assettype
  - topic
  - country
  - subbasin

6 Other Requirements

6.1 Performance Requirements

The system is designed for use through low bandwidth internet connections. In particular, the use of graphical and flash applications is limited.

6.2 Security Requirements

System security is an important aspect: it was established that conX cannot be hacked easily (A 100% percent guarantee cannot be given of course, but so far conX has not been hacked at any time).

All documents are stored in a folder on the webserver which is not directly accessible through the internet. A download-script checks the users access rights to open a document or not. After a successful authorization check the download-script pushes the required document to the browser for download.

This means that documents cannot be accessed through only entering the URL to the document in the webbrowser.

6.3 Screen Design

- Screen resolution starting with 800 pxl * 600 pxl
- NBI identity will be clearly shown
- Basin-wide representative picture on the banner (The picture can be changed from the Back-end)
- Emotions & identity: professional; plain; not overloaded
- Importance of full functionality of the information system (usability & accessibility) on several relevant operating systems and several relevant browsers
- The usage of sound-, animation- and/or video–elements is not needed because of the importance of high performance

7 Future Directions, Hosting and Maintenance

Hosting

The advantages and disadvantages of hosting the system at NBI have been discussed and it was decided that the System will be hosted by hydrophil in Vienna until July 2008. Modalities regarding taking-over the system by NBI still need to be determined as soon as possible. The existing infrastructure of NBI (2 Dual Core 3.2 GHZ, 4GB, 3xHD 146 GB, Raid-5, Backup-Server, Huge USB, Hardware Firewall Cisco ASA5510) fulfils the system requirements for the installation of conX 2.5.

Upload of further information

The Nile-IS currently contains information and data collected during the consultancy (Country Baseline Reports, SVP and SAP project documents) and the Inception Report. The NB countries are asked for their consent to upload further relevant IWRM information.

Possible future developments:

- Multilingualism (Arabic, French, Swahili): System will be prepared at that stage for the implementation of further languages
- Public nile-wiki
- Semantic search, automatic keyword extraction
- Related / similar documents (matched via the meta data structure)
- Social Networking: who knows who, who knows what?

In particular, the addition of web-based GIS mapping functionalities is anticipated for the future.

In the future, the system may also become the nucleus of the database component of the DSS, that will interact with the core database of the DSS through backend-backend communication.

Possible approaches which are to be considered (but which are not included in the standard design of the web-based system as offered here) are

- Offline use of the database with updating when good bandwidth is available (this solution is limited by space restrictions on user's computers)
- Mirroring of the CMS on national servers (as international satellite connections may be the bottleneck).

Appendix E1

Technical Documentation conX (based on Software Framework conX 2.5)

Table of Content

1 About Nile Information System 1

2 About conX 2.5 1

2.1 CMS Basic Features Overview conX 2.5 1
2.1.1 Content Management 1
2.1.2 Rights Management 1
2.1.3 Meta Data 1
2.1.4 Modules 1
2.1.5 conX – Examples of Use 2

3 conX System Architecture 3

4 Technical Specifications conX 2.5 4

4.1 Client System for Back-End Use (Administration View) 4
4.2 Server Requirements conX 2.5 4

5 Database Description 5

5.1 System Tables 5
5.2 Document Type Tables 5
5.3 Relationship Tables 6
5.4 Types in Forms 6
5.5 Interaction of the Database Tables 8
5.6 Document Type Table Description 9
5.7 Configuration Table Description 9
5.8 Relationship Table Description 9
5.9 System Tables and Default Data Types 10
5.10 Files and Images 12
5.11 System Modules and System Messages 13
5.12 Navigation and Meta Data 14

6 Description of the Classes 15

7 Description of important Folders and Files 18

8 System Requirements 19

8.1 Server Requirements conX 2.5 (and therefore Nile Information System) 19

1 About Nile Information System

The vision of Nile-IS is to become a “Knowledge System” in the sense of being “a comprehensive information system on the Nile, which shall support the systematic storage, retrieval, exchange and analysis of relevant information (e.g. documents, data, maps, etc) on water, environment and socio-economic aspects of the Nile Basin”. The system shall provide easy access to and thereby enable the riparians and others to draw from existing knowledge accumulated over the years in numerous projects and studies and contribute to it with information relevant to the sustainable development and management of the water resources of the Nile Basin.

2 About conX 2.5

conX 2.5 is an XML based technology framework for the professional and efficient development of web-based IT solutions as knowledge- & information portals, community environments, project-management tools or company websites.

It is pre-eminent qualified to be used as a Content Management System (CMS) but either to realise community environments, e-learning environments, project management solutions, web-based search engines, etc.

conX uses XML inside of its kernel system!

XML-based applications are standard today. The clear separation of information and design elements, the possibilities of web services and content syndication and the optimum performance of XML are the main advantages of this technology realising web based and mobile solutions.

2.1 CMS Basic Features Overview conX 2.5

2.1.1 Content Management

Content is structured in a tree system (Similar to the well known MS Windows- tree). It includes versioning and multilingual management.

2.1.2 Rights Management

The conX rights management is leaned on the MS Windows rights management structure. All rights can be adapted flexible.

2.1.3 Meta Data

Documents can be enriched with meta data.

2.1.4 Modules

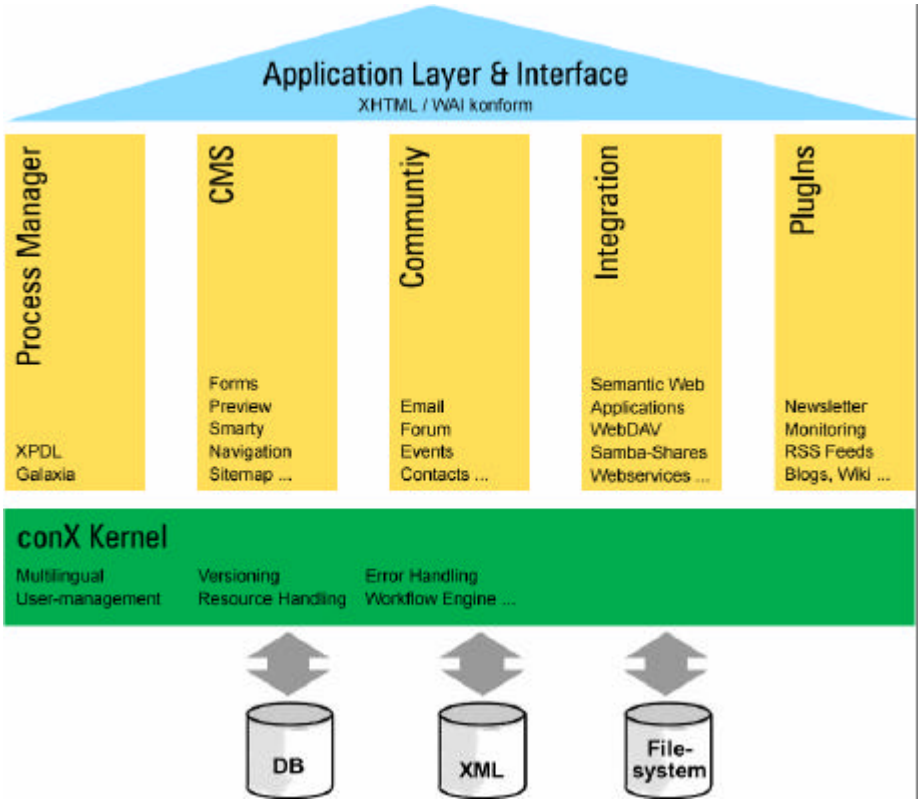
Modules are often adapted third party implementations. For example: RSS reader, link checker, export features, etc.

2.1.5 conX – Examples of Use

- Internet- Information- & and Knowledge-Portals
- Community- and Network Management Systems
- Document Management & Archive Systems
- Seminar Management Systems
- E-Learning Enviroments
- Eventcalendar and Ticketing
- CAWI Systems / e-poll Systems
- Web-based Training & Testing Systems
- XML / PDF Parsers
- Online based search engines

3 conX System Architecture

Figure 3.1 System Architecture conX



4 Technical Specifications conX 2.5

4.1 Client System for Back-End Use (Administration View)

System
Min. Pentium 100 MHz Processor and 64MB RAM Working Memory.
(Corresponds to the minimum requirements of an internet browser)

OS
Windows Win95/NT +, OS with IE 5.5 +

Screen resolution
1024 x 768 +

Internet connection
from Modem-Speed (recommended 56K +)

Browser
MS Internet Explorer 5.5 +

4.2 Server Requirements conX 2.5

Operating System
Unix (e.g. Linux Debian or Red Hat)

Web server
Apache >= 1.3

Middleware
PHP4 (>= 4.1), Java Application Server (JRE1.5)

Database
MySQL (>= 4.0)

Hardware
Min. web server with CPI >500MHz and RAM >256Mb

Comments:
in php.ini globals must be enabled

5 Database Description

The database structure is divided in system tables which are used by conX to store system specific data, document type tables which are used to store data which is managed by users with the conX back end or front end and relationship tables.

5.1 System Tables

- cols
- document
- error
- folder_remember
- logging
- monitoring_search
- objectstack
- rechte
- resource
- styles
- systemmessage
- systemmessageparam
- tree
- treefunction
- treesystem
- treetype

5.2 Document Type Tables

- agency
- agencycategory
- application
- applicationcategory
- appointment
- baseinfo
- bulletin
- casecategories
- casestudies
- content
- dataservice
- dataserviceimage
- doctype
- eligibility
- euwiletter
- euwilettertext
- faq
- faqacp
- faqacpcategory
- folder
- forum
- forumentry
- frontendnavigation
- frontendnavigationdyn

- groupmodule
- help
- image
- knowledge
- land
- link
- linkcategory
- mailalocator
- mailalocatorbulk
- mechanism
- mechanismcategory
- meta
- news
- newsletter
- organisation
- organisationcategory
- paper
- papercategory
- poll
- pollquestion
- project
- symlink
- systemgroup
- systemlanguage
- systemmodule
- systemright
- systemrule
- systemuser
- targetgroup
- theme
- todo
- topic
- usermail

All document type tables also have a configuration table extended with „_conf“. For more information have a look at section „Configuration table description“.

5.3 Relationship Tables

All document type tables can be linked together. This information is stored in the configuration table (type: fid, lku, multimask). All relationships between those document types are stored in relationship tables in the form of „documenttype1_documenttype2“. For more information have a look at section „Relationship table description“.

5.4 Types in Forms

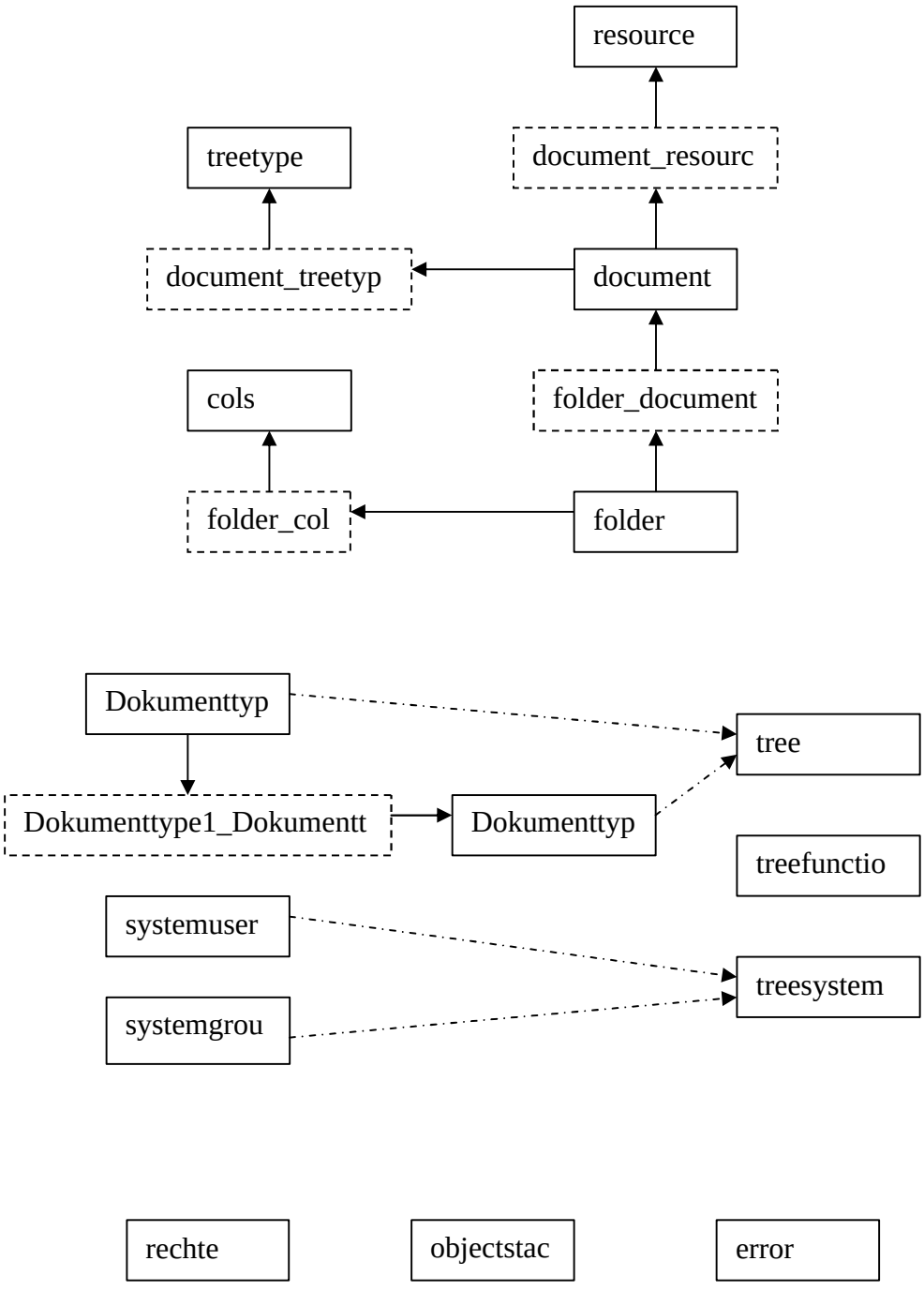
The folling types for inputfields are available.

- text textfield
- link textfield for hyperlinks (incl. regex check)
- email textfield for emai-addresses (incl. regex check)
- int textfield for integers (incl. regex check)
- float textfield for floating point numbers (incl. regex check)

- password — two textfields; the second one is used to check if both are matching
- textarea — textarea field (default: 5 rows)
- datetime — textfield with calendar
- jes_no — radio buttons to choose between yes and no
- dropdown — dropdown field with values from the column „filter“
- internlink — link to one document type entry; selection through the list
- internlink_folder — link to one folder; selection through the list
- fid — link to one document type entry; selection through the list
- lku — link to multiple document type entries; selection through the list
- multimask — link to multiple document type entries; opens a new form
- editor — WYSIWYG text editor field
- image — file upload box for images; add the column „source“ to the data table
- file — file upload box for files; add the column „source“ to the data table

5.5 Interaction of the Database Tables

Figure 5.1 Database Structure conX



5.6 Document Type Table Description

All document type tables have to consist of the following columns:

- id autoincrement identifier
- nav_title the title which is shown in the list
- create_author id of the user who created the document
- change_author id of the last user who modified the document
- create_date date and time when the document was created
- change_date date and time when the document was last modified
- locked username and timestamp of the user who is editing the document
- title required only for multimask-documents

Other fields are used to store document type specific data. If you want to edit this data in the conX backend you have to enable this columns by editing the configuration table.

5.7 Configuration Table Description

To display the form for a document type in the backend it is required that a configuration table exists. The name for this table is the name of the document type table extended with „_conf“.

- id autoincrement identifier
- col unique name of the column in the document type table;
  if it is a link to another document type the name is arbitrary
- title l abel of the field in the form
- position sorting position of the inputfield in the form
- xmltype* unique name of the tag in the xml file which is generated
- type type of the field (eg: text, date, editor); see section „Types in forms“
- resource* name of the document type you want to link to
- rank* numbering if there are more then one link to the same document type
- filter* used for the type „dropdown“: list of options separated with # sign
- dimension* type „float“: number of the decimal places
  type „textarea“: number of the rows (default: 5)
  type „image“ and „file“: file size in byte (default: 5MB)
- needed if set to 1, this field is mandatory (default: 0)
- searchfield if set to 1, this field will be included in the full-text search (default: 0)

*) These fields are used when linking to other document types or when the data is stored as XML.

5.8 Relationship Table Description

All relationships between documents are stored in relationship tables. For example if you want to add to persons (documenttype1) some images (documenttype2) the relationship table has to be named: documenttype1_documenttype2 and the following columns are required:

- fid_documenttype1 id of the entry of the document type1
- fid_documenttype2 id of the entry of the document type2
- rank numbering if there is more than one link to the same document type
- position sorting of the documents in a list

5.9 System Tables and Default Data Types

cols
Contains the column definitions for the list in each folder.

- id auto increment identifier
- col name of the column from the configuration table
- title title of the column shown in the list
- sort defines the default sorting of the list (sort = 1)
- position defines the order of the columns

document
This table describes all documenttypes which are available in conX.

- id auto increment identifier
- name name of the table
- title name which is shown to the user
- to_tree if this value is „1“, it is shown in the tree; default: 0
- to_xml if this value is „1“, it will be a XML file generated; default: 0

document_resource
Relationship table: Here are the relations between documenttypes and resources defined.

document_treetype
Relationship table: Defines in which tree(s) documenttypes are available.

error
This table is used as data store for all mysql errors.

- id auto increment identifier
- date date of the error
- filename name of the file where the error occurs
- message the sql error message
- ip IP address of the causer
- prio priority of the error

folder (folder_conf)
Documenttype: Describes the folders in conX.

- id auto increment identifier
- title title of the folder
- description description of the folder

folder_cols
Relationship table: Here is defined which columns the list in a folder has.

folder_document
Relationship table: Here is defined which documenttype is in a folder allowed.

folder_remember
Relationship table: Contains the position for relations between documenttypes in the tree.

- tabelle table name of the first documenttype
- col table name of related documenttype (see column resource in the configuration table of the first documenttype)
- position the tree_id of the folder where it is stored

objectstack
This is a temporary storage for documents while a user is editing them.

- id auto increment identifier
- sessionid session id of the user
- object serialised object of a document
- fid_user id of the user (table: systemuser)
- lockdate datetime when the object was serialised
- resource resource type of the document
- position the tree_id of the folder where the document is stored
- tabelle name of the documenttype table for this document
- fid_table id of the document
- col column name in the documenttype conf table of lku/fid/multimask
- pathname path name of this document in in the mask
- pathnamelist all pathnames for this documents shown recursive

rechte
Here are the rights for each document stored. The rights are checked in the following order for each document for the logged-in user: 1) User, 2) Group, 3) World

- id auto increment identifier
- fid_id id of the document
- tabelle name of the documenttype table for this document
- user id of the user (table: systemuser)
- group id of the group (table: systemgroup)
- rechte 7-digit number for the rights
- punkt_flag if value is „1“ the document is only shown for user „root“

resource
Here are all available resources stored.

- id auto increment identifier
- name name of the resource in the system
- title name of the resource shown to the users

systemgroup (systemgroup_conf)
Documenttype: Here are the system groups stored which are important for the right handling.

- id autoincrement identifier
- name name of the group
- description description for the group
- systemusers relationship with systemusers

systemuser (systemuser_conf)
Documenttype: Here are the system users stored which are important for the right handling.

- id auto increment identifier
- username login for the user
- email mail address of the user
- firstname first name of the user
- lastname last name of the user
- password password used for the login
- admin_flag if set to 1: user is a system administrator
- logon_flag if set to 1: user is allowed to login in the back end

styles
In this table are available CSS-styles for the back end stored.

- id auto increment identifier
- name name of the style sheet

tree
Entries for the main tree; Here is stored in which folder a document exists.

- id auto increment identifier
- resource name of the document resource
- tabelle name of the documenttype table
- fid_table id of the document
- fid_tree id of the parent entry in the tree table
- frontend_position order in a list (default: NULL)

treefunction
Entries for the function- or moduletree (2nd tree).

- id auto increment identifier
- resource name of the document resource
- tabelle name of the documenttype table
- fid_table id of the document
- fid_tree id of the parent entry in the tree table
- frontend_position order in a list (default: NULL)

treesystem
Entries for the 3rd tree where the users/groups are stored.

- id auto increment identifier
- resource name of the document resource
- tabelle name of the documenttype table
- fid_table id of the document
- fid_tree id of the parent entry in the tree table
- frontend_position order in a list (default: NULL)

treetype
Contains the names of the three tree-types.

- id auto increment identifier
- name name of the table
- title title shown to the user
- last_save date of the last change in the tree

5.10 Files and Images

doctype (doctype_conf)
Contains all file types which are allowed to upload.

- title name of the file type
- shortcut extension of the file (doc, gif, ...)
- type type of the file (image, audio, video, file)

paper (paper_conf)
Contains all files which have the file type file (see doctype).

image (image_conf)
Contains all images with the file type image (see doctype).

5.11 System Modules and System Messages

systemmodule (systemmodule_conf)
Documenttype: Here are all project specific modules stored.

- id auto increment identifier
- title title of the module
- description description of the module
- url relative path where the module is stored in the file system

forum (forum_conf)
Dokumenttype: System module bulletin board

- id auto increment identifier
- title title of the bulletin board
- description description of the bulletin board

forum_cols
Relationship table: Defines which columns are shown for each bulletin board in the list.

forumentry (forumentry_conf)
Documenttype: This table contains all bulletin board entries. This entries are always related to a bulletin board.

- id auto increment identifier
- title title of the entry
- description text of the entry
- name name of the user who wrote the entry
- email email of the user who wrote the entry

forumentry_cols
Relationship table: Defines which columns are shown for a bulletin board entry.

systemmessage
Here are messages stored which a user sent to other users. in addition it is also possible to send a message via mail.

- id auto increment identifier
- sender user id of the sender
- addressee user id to the recipient
- messagetype type of the message
- subject subject of the message
- message text of the message

systemmessageparam
Depending on the message type it is possible to store additional parameter for a message here.

- id auto increment identifier
- fid_systemmessage message id
- key_param name of the parameter
- value_param value of the parameter

todo (todo_conf/todo_fconf)
Documenttype: In this table are to dos stored.

5.12 Navigation and Meta Data

frontendnavigation (frontendnavigation_conf)
Here are all navigation elements stored.

- id auto increment identifier
- title name of the hyper link
- image image instead of the a text link if desired
- fid_tree tree id of a document
- fid_tree_list tree id of a folder
- fid_treetype treetype id where the document or folder is stored
- path path of a script which should be used
- fix if set to 1, the sub navigation items are shown

frontendnavigation_frontendnavigation
Relationship table: This table is used to store the hierarchical structure of the navigation.

frontendnavigationdyn (frontendnavigationdyn_conf)
This table is used for dynamic navigation creation with scripts.

meta (meta_conf)
Documenttype: Here are the HTML-title and meta tags stored.

- id auto increment identifier
- defaulttitle default HTML-title
- publisher meta tag for the HTML-header
- copyright meta tag for the HTML-header
- language meta tag for the HTML-header
- robots meta tag for the HTML-header

6 Description of the Classes

Resource
With this class you can edit all common documenttype data.

- create a document
- load available documents
- lock a document for other user (while editing)
- release a document for other user
- read the data from the document
- update a document
- save a document
- delete a document
- move a document from one folder to an other
- copy a document from one folder to an other

liste
This class is used to show documents in a folder as list. In addition you can also manage with this class which columns should be shown in a specific list.

- read the columns
- sort the list according to a specific column (asc/desc)
- add new columns
- remove columns
- get the rows (documents) for the folder
- add a new document to a folder
- delete a document from the folder
- move the document in the list up or down

right
With this class it is possible to manage the rights for a specific document and user/group/world.

- release/lock a document for the world (every user)
- check the if a document is released or not
- read and set specific rights
- read all rights
- set default values when saving a new document
- remove rights when a document was deleted
- get user who have rights for a documents
- get groups who have rights for a document
- copy the rights from one document to an other one

db
This class is used to connect to a database, execute queries and do database operations.

- open a connection to a database
- close a connection
- query the database
- get the result set
- list the results
- check if a specific column exists
- get the last inserted id

user
The user class is used to manage and authenticate a user in conX and the front end.

- load all data of a user
- set the system language referring to the user preferences
- authenticate the user with username and password
- set a new password
- update the last login time
- read the groups a user belongs to
- transfer the rights for documents to an other user (when deleting a user)

error
This class writes error messages in the database (table: error) or in a log file.

session
This class is responsible for handling session data for a logged-in user.

stack
The stack class handles the objectstack. This class is used by other classes to save serialized objects of a document temporary in the database while a user is editing a it.

tree
The tree.class is responsible to read the data from a specific tree and build the tree-navigation menu items.

validator
This class is used to check the validity of input data from a user.

xml
This class is used to store the data also as XML file when a document is saved.

mysmarty
This class configures and extends the smarty template engine.

smarty
The smarty template engine package.

cis_mask, cis_mask2, cis_mask3, cis_mask4
The cis_mask classes are used to display different user interface elements in the CIS environment.

cis_usermail
The cis_usermail class handles all internal sent messages.

cis_user
This class handles the login procedures from the front end.

cis_whatsnew
This class is responsible to show after a login new events, documents, messages, users, groups and bulletin board messages to the user.

baseinfo, eventsinfo, faqinfo, knowledgeinfo, linksinfo, metainformation, newsinfo, linksinfo, participantsinfo
This classes are used to handle different documenttypes as class.

participantsinfo_list
This class generates a list of participants for events.

primenav, secondnav, thirdnav
This classes are used to get the navigation elements for the different navigation levels.

eventsbox, filterbox, newsbox, participantsbox, toolbox
This classes are used to display context information in the different boxes on the website.

htmlmimemail
This class is used to send e-mail messages to standard e-mail addresses.

vcard
The vcard class is used to generate vCard files out of user data.

rsswriter
The rsswriter class is used to generate RSS feeds.

ical
This class generates iCalendar files for appointments so user can add them to their Calendar application.

7 Description of important Folders and Files

Permissions
All public accessible files must have read rights for the web server user.
The folders “file_upload” and “templates_c” require read and write rights for the web server user.

inc/
This folder contains the main configuration file and common functions which are often used in differnt parts of the system.

inc/conf.php
The file conf.php contains the most important system settings.

- PROJECT_TITLE the project title (use only chars)
- HOME_LINK public url to the root system
- DB_HOST the database server host
- DB_USER the database username
- DB_PASS the database password
- DB_DB the database name
- BASE_FOLDER path in the file system

inc/functions.php
This file contains global functions like formatting date or time, render links, get the home folder for a user and display HTML output for different documenttypes like baseinfo, knowledgeinfo, participants, events, news, faq and so on.
There are also functions to generate the lists for news, events or the site map.

inc/cis*.php
The files cisdocuments.php, cisforum.php, cisgroup.php and cisusers.php contains functions to lookup and display documents, user details, groups with their users and bulletin boards.

index.php
The index.php file acts as main controller and renders the content for the website.

admin/
In this folder the conX back end resides. Normally it is not necessary to change here something.

cis/
In the cis folder there are all files which are used for CIS system.

classes/
Here are all above mentioned classes stored.

file_upload/
Here are all files and images stored, which the users have uploaded through the web interface.

functions/
In the functions folder there are the files needed for the after save function to add metainformation.

modules/
Special modules like the CSV export, the EUWI-letter and monitoring are available in this folder.

8 System Requirements

The existing Nile Information System is running on the software framework conX, version 2.5 using a LAMP architecture:

- Linux
- Apache Web server
- MySQL database
- PHP as coding language

8.1 Server Requirements conX 2.5 (and therefore Nile Information System)

Web server:
Apache web server >= 1.3.26

- PHP4 ? 4.1.2-7
- PHP4-gd
- PHP4-MySQL
- PHP4-XML
- gettext 0.10.40-5

Database:

- MYSQL ? 4.0.24

Further needed:
In php.ini globals must be ENABLED

Server in use at the moment

- Web server
  IBM eServer xSeries 306m / rackmountable, 1U, 1x Intel 3000 MHz., 1GB RAM, Serial attached SCSI integrated, 1x 73,4GB HDD SAS hot swap, CD-ROM, Ethernet 10/100/1000B-TX

- Database server
  IBM eserver xSeries x226 Modell 8648; Tower - 1 x Xeon 3,0 GHz (Dual Core integrated); RAM 2GB, 1 x 80GB, CD, LAN EN, Fast EN, Gigabit EN

- Also in Use:
  - Backup (Server) system – daily back-up / additional weekly back-up: web-server & database
  - USV – emergency power; System stability - monitoring
  - Web-log statistics for the administration included in use: WebLogExpert (See: http://www.weblogexpert.com)

Appendix E2

Asset Type Specification

Asset Type: Themes

Field Name	*Oblig.*	*Field Type*	*Comment*
Theme or Sector	*	Single-line text field	
Country	*	1:1 Relation Country **A**	
Description		Text area	

Key Facts	*Title*	Free to chose
	Description	Text area
	Picture	Relation: 1:1 Picture **A**
	Documents	Relation: 1:m Documents **A**

Policy / Legal Framework	*Title*	Title Dropdown: Policy# Strategy# Legal Situation# Regulation# International Treaties
	Description	Text area
	Documents	Relation: 1:m Documents **A**
	Treaties	Relation: 1:m Treaties **P**

Institutional Framework	*Title*	Title Dropdown: Policy/Strategy Formulation# Governance# Regulation# Sector Investment# Service Provision# Sector Coordination# Sector Reform# Regional Cooperation# Development Partners# Research & Education# Data Collection# Civil Society# Private Sector# Community Level
	Description	Text area
	Documents	Relation: 1:m Documents **A**
	Institutions	Relation: 1:m Institutions **A**
	Projects	Relation: 1:m Projects **A**

Issues and Constraints	*Title*	Title Dropdown: Key Issues# Conflicts# Potentials# Innovative Approaches# Climate Change# Funding# Cost Recovery#
	Description	Text area
	Documents	Relation: 1:m Documents **A**
	Projects	Relation: 1:m Projects **A**

Infrastructure	*Title*	Free to chose
	Description	Text area
	Picture	Relation: 1:1 Picture
	Infrastructure	Relation: 1:m Infrastructure **P**

Data Availability	*Title*	Title Dropdown: Databases# Time Series Data# Spatial Data# Identified Gaps#
	Description	Text area
	Databases	Relation: 1:m Databases **P**
	Time Series Data	Relation: 1:m Time Series Data **P**
	Spatial Data	Relation: 1:m Spatial Data **P**
	Documents	Relation: 1:m Documents **A**

Models and Indicators	*Titel*	Title Dropdown: Models# Indicators# Comment#
	Description	Text area
	Models	Relation 1:m Models **P**
	Documents	Relation: 1:m Documents **A**

Entry By	Automated by system
Entry Date	Automated by system

Asset Type: Institution

Field Name	*Oblig.*	*Field Type*	*Comment*
Institution Name	*	Single-line text field	
Acronym		Single-line text field	
Full name		Text area	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-Basin		1:m Relation Sub-basin **A**	
Thematical Focus	*	Text area	
Territorial Scope	*	Drop down list	national# sub-national,# international/regional
Type of Organisation	*	Drop down list	Government/administration# Parastatal/public Authority# International Organisation# Research/educational institution# Private Sector# NGO# Voluntary Association
Website		URL	

Organisational Structure	*Title*	Drop down list: Legal Status# Steering# Internal Organisation# Organisation Chart# Decentralisation# Changes# Performance, M&E# Stakeholder Participation
	Description	Text area
	Picture	Relation: 1:1 Picture **A**
	Documents	Relation: 1:m Documents **A**

Mission and Activities	*Title*	Drop down list: Mission / Responsibilities# Actual Functions# Goals & Strategy# Major Programmes# Services provided
	Description	Text area
	Documents	Relation: 1:m Documents **A**

Finance	*Title*	Drop down list: Budget# Revenue# Financial Supervision#
	Description	Text area
	Documents	Relation: 1:m Documents **A**

Human Resources	*Title*	Drop down list: Staff Number# Staff Qualification# HR Management# Training Initiatives# Pay Scale
	Description	Text area
	Documents	Relation: 1:m Documents **A**

IT Capacities	*Title*	Drop down list: IT/GIS staff capacities# IT/GIS infrastructure# Internet Access
	Description	Text area
	Documents	Relation: 1:m Documents **A**

NBI-DSS	*Title*	Drop down list: Scenario Definition# Decision Making# Interest in NBI-DSS# Information Needs
	Description	Text area
	Documents	Relation: 1:m Documents **A**

Data collection and Models	*Title*	Drop down list: Time Series Data / Statistical Data# Spatial Data (GIS/maps)# Databases# Models# Identified Gaps
	Description	Text area
	Documents	Relation: 1:m Documents **A**
	Databases	Relation: 1:m Databases **P**
	Time Series Data	Relation: 1:m Time Series Data **P**
	Spatial Data	Relation: 1:m Spatial Data **P**
	Models	Relation: 1:m Models **P**

Field Name	*Oblig.*	*Field Type*	*Comment*
Institutional Cooperation		Text area	
Institutions		1:m Relation Institutions **A**	
Participation in international projects		Text area	
Projects		1:m Relation Projects **A**	

Field Name	*Oblig.*	*Field Type*	*Comment*
Contact Details		Text area	

Entry by		Automated by system	
Date		Automated by system	

Asset Type: Country

Field Name	*Oblig.*	*Field Type*	*Comment*
Country	*	Single-line text field	
ISO Code	*	Single-line text field	
Area		Single-line text field	
Area Nile Basin		Single-line text field	
Area Percent Nile Basin		Single-line text field	
Population		Single-line text field	
Population Percent Nile Basin		Single-line text field	

Asset Type: Sub-basin

Field Name	*Oblig.*	*Field Type*	*Comment*
Sub-basin	*	Single-line text field	
SB ID	*	Single-line text field	
SB Area		Single-line text field	
SB Population		Single-line text field	
Upstream Sub-basin	*	1:m Relation Sub-basin **A**	
Downstream Sub-basin	*	1:m Relation Sub-basin **A**	
Country	*	1:m Relation Country **A**	

Asset Type: Infrastructure

Field Name	*Oblig.*	*Field Type*	*Comment*
Infrastructure Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin	*	1:m Relation Sub-basin **A**	
Province		Single-line text field	
Location/coordinates		Single-line text field	
Status		Drop down list: planned# under construction# existing# out of use#	
Construction Year		Single-line text field	
Owner/operator		Text area	
Characteristics		Text area	
Water Rights		Text area	
Water Requirements		Text area	
Source of Water		Single-line text field	
Wastewater discharge to		Single-line text field	
Data used		1:m Relation Time Series Data **A**	
Documents		1:m Relation Documents **A**	
Information Needs		Text area	
Contact Details		Text area	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Project

Field Name	*Oblig.*	*Field Type*	*Comment*
Project Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Acronym		Single-line text field	
Lead/host		1:1 Relation Institution **A**	
Member Institutions		1:m Relation Institution **P**	
Start Year		Single-line text field	
Funding		Text area	
Status		Dropdown: Planned# Ongoing# Terminated	
Objectives		Text area	
Activities		Text area	
Achievements/results		Text area	
Information Needs		Text area	
Databases		1:m Relation Database **A**	
Models		1:m Relation Model **A**	
Documents		1:m Relation Documents **A**	
Website		URL	
Contact Details		Text area	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Treaty

Field Name	*Oblig.*	*Field Type*	*Comment*
Treaty Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Full Title		Single-line text field	
Year		Single-line text field	
Legal Status		Text area	
Key Contents		Text area	
Data Requirements		1:m Relation Time Series Data **A**	
Documents		1:m Relation Documents **A**	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Database

Field Name	*Oblig.*	*Field Type*	*Comment*
Database Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Full Name		Single-line text field	
Content		Text area	
Purpose		Text area	
Region		Text area	
Year		Single-line text field	
Last Modification		Single-line text field	
Developed by		Text area	
Institution		1:m Relation Institutions **A**	
Host Institution		1:m Relation Institutions **A**	
Documents		1:m Relation Documents **A**	
Time Series Data		1:m Relation Time Series Data **P**	
Spatial Data		1:m Relation Spatial Data **P**	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Time Series Data

Field Name	*Oblig.*	*Field Type*	*Comment*
Dataset Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Parameter(s)		Text area	
Unit(s)		Single-line text field	
Data Origin		Text area	
Spatial Resolution		Text area	
Time Resolution		Text area	
Method(s)		Text area	
Start Date		Single-line text field	
End Date		Single-line text field	
Completeness		Text area	
Data Quality		Text area	
Data Transmission		Text area	
Data Storage		1:m Relation Databases **A**	
Data Format		Single-line text field	
Data Accessibility		Drop down list: On the internet# Published in yearbook# On request# Restricted use only#	
Responsible for Update		1:m Relation Institutions **A**	
Data Contact		Text area	
Documents		1:m Relation Documents **A**	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Spatial Data

Field Name	*Oblig.*	*Field Type*	*Comment*
Dataset Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Content		Text area	
Data Origin		Text area	
Spatial Resolution		Text area	
Reference Grid		Single-line text field	
Year		Single-line text field	
Last Update		Single-line text field	
Data Quality		Text area	
Data Storage		1:m Relation Databases **A**	
Data Format		Single-line text field	
Data Accessibility		Drop down list: On the internet# Published in yearbook# On request# Restricted use only	
Responsible for Update		1:m Relation Institutions **A**	
Data Contact		Text area	
Documents		1m Relation Documents **A**	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Model

Field Name	*Oblig.*	*Field Type*	*Comment*
Model Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Full Name		Text Area	
Type		Text area	
Purpose		Text area	
Region		Text area	
Spatial Resolution		Single-line text field	
Time Resolution		Single-line text field	
Input Parameters		Text area	
Output Parameters		Text area	
Year		Single-line text field	
Developed by		Text area	
Institution		1:m Relation Institutions	
Last Modification		Single-line text field	
Programme Code		Text area	
Status		Drop-down: Under development# In operational use# Used for educational Purposes# Runs but not used# Not working# Working but no capacities to use it#	
Host Institution		1:m Relation Institutions **A**	
Documents		1:m Relation Documents **A**	
Entry by		Automated by system	
Date		Automated by system	

Asset Type: Document

Field Name	*Oblig.*	*Field Type*	*DC Core (Metadata)*
Document Name	*	Single-line text field	
Theme	*	1:m Relation Theme **A**	
Country	*	1:1 Relation Country **A**	
Sub-basin		1:m Relation Sub-basin **A**	
Type of Document		Drop down: Policy/strategy paper# Scientific study# Consultancy report# Feasibility study#	
Full Title		Text area	DC: Title
Editor		Text area	DC: publisher
Institution		1:m Relation Institutions **A**	
Author (Person)		Single-line text field	DC: creator
Publication Year		Single-line text field	DC: date
Language		Drop down: English# French# Arabic# Other	DC: language
Abstract		Text area	DC: description
Availability		Drop down list: On the internet# On request# Restricted use only	DC: rights
Document Location		URL	DC: identifier
Document Format		Drop-down: *.doc# *.pdf# Hardcopy#	
Entry by		Automated by system	
Date		Automated by system	
Document		File	

Asset Type: Basic Information

Field Name	*Oblig.*	*Field Type*	*Comment*
Title	*	Single-line text field	
Lead		Text area	
Theme		1:m Relation Theme	
Country		1:1 Relation Country	
Sub-basin		1:m Relation Sub-basin	
Content Blocks		See content block	
Subtitle		Single-line text field	
Text		Text area	
Opt. Picture		1:m Relation Picture	
Opt. Document		1:m Relation Document	
Opt. Time Series Data		1:m Relation Time Series Data	
Opt. Spatial Data		1:m Relation Spatial Data	
Opt. Database		1:m Relation Databases	
Opt. Models		1:m Relation Models	

Asset Type: Picture

Field Name	*Oblig.*	*Field Type*	*Comment*
Title	*	Single-line text field	
Picture	*	Image	

Asset Type: Users

Field Name	*Oblig.*	*Field Type*	*Comment*
First Name	*	Single-line text field	
Last Name	*	Single-line text field	
Email	*	Single-line text field	
Institution	*	Single-line text field	
Language preference		Drop down: English# French#	
Job Title / Position		Single-line text field	
Change Password		Single-line text field	
Confirm Password		Single-line text field	
Administrator		Radio Button(Yes/No)	
Back-End Access		Radio Button(Yes/No)	
Group		1:m Relation Group **A**	
Leader in Group		1:m Relation Group **A**	

Asset Type: Group

Field Name	*Oblig.*	*Field Type*	*Comment*
Title		Single-line text field	
Description		Text area	
Members		1:m Relation Users **P**	
Leaders		1:m Relation Users **P**	

Note:

Relations marked with A: (active link)

- The user can associate an asset with another asset. The associates will be shown at the front-end.

Relations marked with P: (passive link)

- The user can not associate with an asset. The associates will we shown inverted at the front-end from the associated asset.

Appendix E3

Process of Development

Development Steps:

1. Preliminary description of system requirements reported by Reinold Seidelmann on 30 July, 2007
2. System requirements meeting held at NBI in Addis Abeba, Wednesday 8 August and Thursday, 9 August 2007
3. 12th August 2007: summary on meeting
4. 15th August 2007: send an agreement on exclusive use of conX for the KN-IS
5. 17th August 2007: contacted by Andreas Koller (Virtual Server Access, Source Code)
6. 27th August 2007: handing over conX 2.5 with documentation and installation instruction
7. 30th August 2007: set up first prototype of the Nile-IS V0.1 http://nbi.hydrophil.at
8. 16th September 2007: CRS V0.99, prototype 0.6 beta
9. 23rd September 2007: CRS V1.0 beta, prototype 0.8 beta
10. 30th September 2007: CRS V1.0 stable is confirmed, Prototype 1.0 beta online
11. 1st October 2007: CRS (excerpt) will be put into inception report
12. 18th and 19th October 2007: inception Workshop Addis Abeba
13. November 2007: refinement Asset Types
14. December 2007: set up development enviroment http://devnbi.hydrophil.at for testing purposes
15. November 2007 - January 2008: developing and testing
16. 10 January 2008: Nile-IS online Release Candidate 1
17. 25 January 2008: Nile-IS online Stable Version Online