

Bericht Nr. 2996.1

**Eine E-Commerce Plattform für Ingenieurleistungen im
Schiffbau
- Anleitung für den Administrator -**

Teilbericht zum FuE-Verbundvorhaben

„Entwicklung einer E-Commerce-Plattform für Ingenieurleistungen im Schiffbau“

Potsdam, August 2004



**Entwicklung einer E-Commerce-Plattform für Ingenieurleistungen im Schiffbau
- Anleitung für den Administrator -**

Auftraggeber: Schiffbau-Versuchsanstalt Potsdam GmbH
Marquardter Chaussee 100
14469 Potsdam
Tel. (0331) 567120, Fax (0331) 5671249

FuE-Vorhaben: Entwicklung einer E-Commerce Plattform für Ingenieurleistungen im
Schiffbau

Förderung: BMBF

Förderkennzeichen: 18S0216

Bearbeiter: Dipl.-Inf. F. Domeyer (Universität Potsdam)
Dipl.-Math. H. Wuttke

Dieser Bericht enthält: 35 Blatt Text
5 Abbildungen

Potsdam, 20.August 2004

Inhaltsverzeichnis

	Inhaltsverzeichnis	3
1.	Einführung	5
1.1	Startseite ePing	6
1.2	Systemvoraussetzungen	7
1.3	Notwendige Komponenten	7
1.4	Installation der erforderlichen Voraussetzungen	7
1.5	Installationsanleitung der Dienstplattform ePING	9
1.5.1	Installationsprozedur	9
1.5.2	Manuelle Anpassungen	10
1.5.3	Sicherheitsanpassung	11
1.5.3.1	Administrator von ePING	11
1.5.3.2	Administrator von phpMyAdmin	11
1.6	Deinstallation	12
1.7	Typische Fehlermeldungen	12
2	Wartung	12
2.1	Lauffähigkeit des Systems	12
2.2	Administrator	12
2.3	Mitarbeiter	13
3	Aufbau der Dienstleistungsplattform ePING	13
3.1	Aufbau der internen Struktur von ePing (Verzeichnisse)	13
3.2	Aufbau des öffentlichen und gesicherten Bereiches	15
3.2.1	Bereiche ePing, Conditions of Use, About ePing	15
3.2.2	Bereich Free Area	15
3.2.3	Bereich Registered Area	15
3.2.4	Bereich Individual Area	15
3.2.5	Bereich Mitarbeiter	16
3.2.6	Bereich Admin(istrator)	16
4	Steuerung der Dienstleistungsplattform durch PHP	16
4.1	Usergruppen einrichten	18
4.2	Rechenverfahren einzelnen Usergruppen zuordnen	19
5	Datenhaltung	20
6	Anleitung zur Integration von Rechenverfahren	20
6.1	Anforderung an die Rechenverfahren	20
6.1.1	Notwendige Dateien	21
6.1.2	Arbeitsschritte zum Einbinden eines Rechenverfahren	22
6.2	Einbinden eines externen Programms (XML)	22
6.2.1	Vorarbeiten	22
6.2.2	Erstellen der XSL-Datei	23
6.2.2.1	Anforderungen an die XML-Struktur	23

6.2.2.2	Die XSL-Datei	24
6.2.2.3	Ausgabe XSL-Datei	25
6.2.2.4	Ausgabe als reine Textdatei einbinden (Non-XML-Ausgabe)	26
6.2.2.5	Andere XSL-Sprach-Dateien	26
6.2.2.6	Anpassung der XSL bei Aufteilung der Eingabedaten über mehrere Seiten	26
6.2.3	Erstellen der PHP - Funktionalität des Rechenverfahren	26
6.2.3.1	Holtrop.php	27
6.2.3.2	Holtrop_main.php	28
6.2.3.3	Holtrop_error_table.phtml	29
6.2.3.4	Bilder anzeigen	29
6.2.3.5	Die Schnittstelle zum externen Rechenverfahren	30
6.2.4	Einbinden eines externen Programms (Non-XML)	31
6.2.5	Erweiterung für das nicht interaktive Ansprechen von Rechenverfahren	32
6.2.6	Einbinden von Rechenverfahren in den individuellen Bereich	32
6.2.7	Änderungen für das Einbinden von Rechenverfahren in den öffentlichen Bereich	33
6.2.8	Anpassungen bei Rechenverfahren mit sehr langer Ausführungszeit	33
7	Individueller Bereich	33
7.1	Einstellen von Inhalten	34
7.2	Auf die Inhalte zugreifen	35
8	Sicherheit	35
8.1.1	Schutz der Daten	35
8.1.2	Schutz bei der Datenübermittlung	35

Abbildungsverzeichnis

Abbildung 1: Startseite	6
Abbildung 2: Wahl des Zielverzeichnisses	9
Abbildung 3: Eingabe aller notwendigen Parameter	9
Abbildung 4: Dateistruktur von ePing	13
Abbildung 5: Einstellen von Inhalten in den individuellen Bereich	34

1. Einführung

Die E-Commerce-Plattform für Ingenieurleistungen im Schiffbau (ePING) ist mit dem Ziel entwickelt worden, ein Client-Server-System zu erstellen, das zur projektbezogenen Kommunikation, Beauftragung, Abwicklung und Abrechnung von Dienstleistungen verwendet werden kann.

Diese technische Dokumentation beschreibt die Voraussetzungen, die Installation, das Betreiben und das Erweitern der Plattform und ist als Arbeitsgrundlage für den Administrator gedacht.

Die hardwareseitigen Anforderungen an den Server sind abhängig von der Häufigkeit der zu erwartenden Anfragen und der Rechenzeit der installierten Programme. Günstig ist ein Mehrprozessor-Rechner mit einem Server-Betriebssystem, falls dauerhaft gleichzeitig mehrere Anfragen zu erwarten sind.

Die notwendigen Komponenten, die die Plattform voraussetzt, wurden so gewählt, dass weit verbreitete und geläufige Werkzeuge zum Einsatz kommen, um den Einarbeitungs- und Betreuungsaufwand für den Administrator gering zu halten. Zusätzlich wurden die Werkzeuge so abgestimmt, dass der Aufwand für das Portieren auf ein anderes Betriebssystem gering bleibt. Die Schnittstellen zwischen den einzelnen Komponenten sind so ausgelegt, dass auch andere Web-Server oder andere Datenbanken eingebunden werden können.

Die Installation des E-Commerce-Systems setzt das fehlerfreie Funktionieren der notwendigen Komponenten voraus. Das Installationsprogramm für die Dienstplattform gestattet das Einrichten der erforderlichen Verzeichnisse und die Bereitstellung der notwendigen Programme für den Standardanwendungsfall. Die manuellen Anpassungen, soweit sie erforderlich sind, werden ausführlich beschrieben.

Der Betrieb der Dienstleistungs-Plattform wird durch die mitgelieferten Werkzeuge sichergestellt. Es wird eine abgestufte Hierarchie von Nutzern verwaltet, die entsprechend ihrer eingeräumten Rechte eine unterschiedliche Sicht auf das Gesamtsystem besitzen. Aufgabe des Administrators ist es, diese Nutzerstruktur zu verwalten und den Gegebenheiten des geplanten Betriebes der Plattform anzupassen.

Kern der Plattform ist die Bereitstellung von Rechenverfahren für die einzelnen Nutzer. Der Datenaustausch mit diesen Verfahren wurde überwiegend über XML realisiert und wird für das Ergänzen um weitere Verfahren auch empfohlen. Die Dokumentation beschreibt die Anforderungen, die an die Verfahren gestellt werden, um sie möglichst einfach integrieren zu können und sicher zu stellen, dass sie im normalen Dauerbetrieb auch störungsfrei arbeiten. Schwerpunkt liegt dabei auf dem Prüfen aller Eingabedaten und der Vermeidung unkontrollierter Programmzustände.

Wesentlicher Bestandteil der Plattform sind die verwendeten Sicherheitsmechanismen. Sie entsprechen den geltenden Standards, sollten jedoch unbedingt dem Fortschreiten der technischen Entwicklung angepasst werden.

1.1 Startseite ePing

Nach der erfolgreichen Installation sollte die Startseite von ePING im Browser wie folgt aussehen:

The screenshot shows the homepage of the ePING website. At the top, there is a header banner with a blue and white design, featuring the SVA ePING logo on the right and a search bar with the text 'Please select QUICKFINDER'. Below the banner is a navigation menu with a yellow highlight on 'ePing' and other options like 'Free Area', 'Registered Area', 'Individual Area', 'Conditions Of Use', 'About ePING', and 'SVA'. The main content area is titled 'ePING - Engineering Assistance in Hydrodynamics'. It starts with a welcome message: 'This online service is brought to you by > Schiffbau-Versuchsanstalt Potsdam GmbH (SVA)'. Below this is a section titled 'What is ePING?' with three bullet points: '- ePING is our new and growing web site which helps to assists you in the field of hydrodynamics.', '- ePING is a joint research project together with > University of Potsdam.', and '- The project is funded by BMBF, the German 'Bundesministerium für Bildung und Forschung''. The next section is 'ePing offers 3 different working levels' with links to 'Free area', 'Registered area', 'Individual area', and 'Further informations about the offered methods'. The 'Free area' section states: 'Using this area is free of charge. The service is provided without any garantie and all possible warranties are disclaimed. Feel free to test the functions we offer - our staff is using the ePING platform in their engineering work as well. And do not forget: we need your feedback in order to improve this service.' The 'Registered area' section says: 'Just register online to gain access to more services in the registered area. This sites are encrypted with the latest encryption technology. Additionally will the SVA Potsdam ensure that your data is secure and can only be accessed or altered by you or by members of the SVA staff.' The 'Individual area' section mentions: 'In the individual area you can gain exclusive access to further applications and/or specific information. For this you need additional agreements or projects with the SVA.' At the bottom, it says: 'If you have any questions contact > Hans Wuttke'. The footer of the page reads 'styled by CultVision'.

Abbildung 1: Startseite

Die Dienstleistungsplattform ePING ist durch das Verwenden der Programmiersprache PHP auf jedes Betriebssystem portierbar, doch wird im folgenden davon ausgegangen, dass ePING auf dem Zielsystem Windows (2000, XP) installiert werden soll. Damit ePING korrekt läuft, muss sowohl der Server, als auch der Client mit Cookies umgehen können. Auf der Serverseite muss diese Eigenschaft in PHP aktiviert werden.

1.2 Systemvoraussetzungen

Bevor ePING installiert wird, sollte sichergestellt werden, dass der Computer ungefähr den folgenden Anforderungen entspricht und die im nächsten Kapitel beschriebenen Komponenten vorliegen.

Minimalkonfiguration

- 128 MB Arbeitsspeicher
- 1,x GHz Prozessor
- 50 MB freien Festplattenspeicher

Empfohlene Konfiguration

(anzupassen, wenn eines der eingebundenen Rechenverfahren höhere Anforderungen stellt)

- Windows XP/2000 Server
- 512 MB Arbeitsspeicher
- 2,x GHz Prozessor
- 200 MB freien Festplattenspeicher

1.3 Notwendige Komponenten

Die Installation dieser Komponenten wird im folgenden Kapitel beschrieben. Hier zunächst die Liste der notwendigen Programme:

- Apache-Server (oder ein anderer Web-Server)
- MySQL-Datenbank (oder eine andere Datenbank)
- PHP 4.x (oder neuere Version)
- GhostScript (notwendig für die Umwandlung von PS-Dateien)
- Optional ein Bildbearbeitungsprogramm
(Es wird intern IrfanView für Windows installiert und verwendet)
- XML-Validator (wird mitgeliefert)

1.4 Installation der erforderlichen Voraussetzungen

1. Installation von PHP [nach c:\php]

- Ausführen von php_*_installer.exe

- Bei der Installation, muss der SMTP-Server eingetragen werden, um die Mail-Funktion freizuschalten (wird von ePING benutzt), weiterhin ist die From-Mail-Adresse einzutragen [eping@]
 - Auswahl von Apache als Web-Server (hat keinen Einfluss bei der Installation)
- Auspacken von php*_win32.zip nach c:\php und dabei einfach die schon vorhandenen Dateien überschreiben
- Editieren der vom Installer angelegten php.ini Datei (c:\windows\php.ini)
 - Setzen von extension_dir auf c:/php/extensions
 - Aktivieren der Extension
 - php_domxml.dll
 - php_iconv.dll
 - php_xslt.dllin der php.ini entfernen von ';' vor dem Eintrag (wenn nicht vorhanden, nachtragen)
- Es sollte darauf geachtet werden, dass folgende Einträge in der php.ini (wenn sie überhaupt vorhanden sind) wie aufgeführt gesetzt sind:
 - short_open_tag = ON
 - file_uploads = ON
 - session.use_cookies = 1
 - session.cookie_path = / (bzw auf ein existierens Verzeichnis setzen)
 - error_reporting = E_ALL & ~E_NOTICE
- Weiterhin können folgende Einträge angepasst werden:
 - log_errors = ON
 - display_errors = OFF
- Kopieren von c:\php\dll* nach c:\windows\system32 oder alternativ c:\php\dll in der PATH-Umgebungsvariable eintragen
- Kopieren von c:\php\php4ts.dll nach c:\windows\system32 oder alternativ c:\php in der PATH-Umgebungsvariable eintragen (notwendig für Apache mit Php4 als Module)
- testen der Lauffähigkeit von php durch Aufruf von c:\php\php.exe

2. Installation von Apache 2 [nach c:\programme]

- Ausführen von Apache_*.msi
- Einbinden von PHP in Apache
 - Editieren von httpd.conf (c:\programme\apache group\apache2\conf\httpd.conf)
 - Hinzufügen von

```
LoadModule php4_module "c:/php/sapi/php4apache2.dll"
AddType application/x-httpd-php .php .phtml .php3 .inc
```
- Apache neu starten

3. Installation von Ghostscript [nach c:\gs]

- Ausführen von gs*w32.exe (Bsp: gs811w32.exe)

4. Installation von MySQL [nach c:\mysql]

- Ausführen von setup.exe
- Starten von c:\mysql\bin\winmysqladmin.exe (Der dort einzurichtende User hat nur etwas mit WinMySQLAdmin zu tun)
- Einrichten eines neuen lokalen User in der Datenbank oder zumindest den vorhandenen User ‚root‘ mit einem Passwort absichern. [Zugriff auf der Kommandozeile (root ohne

Passwort) per `c:\mysql\bin\mysql.exe -h localhost -u root` – dann in der Datenbank mysql in der Tabelle user die Änderungen vornehmen - Bsp: root Passwort: mypwd01]

- Gegebenenfalls ist die Datenbank nach außen abzusichern. Dabei ist darauf zu achten, dass es mindestens einen User gibt, welcher lokale Zugriffsrechte auf alle MySQL-Befehle hat und bei ePING genutzt wird.
- Gegebenenfalls den Rechner neu starten, damit die Datenbank läuft.
- Testen der Datenbank [`:\mysql\bin\mysql.exe -h localhost -u root -pPASSWORT`]
Achtung: hinter dem Flag `-p` ist KEIN Leerzeichen!

1.5 Installationsanleitung der Dienstplattform ePING

1.5.1 Installationsprozedur

Installation von ePING [nach `c:\eping`]

- Ausführen von `eping_setup.exe` und ausfüllen der geforderten Angaben (siehe Bilder)

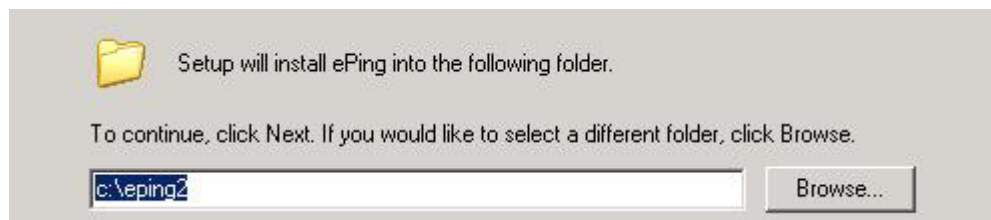


Abbildung 2: Wahl des Zielverzeichnisses

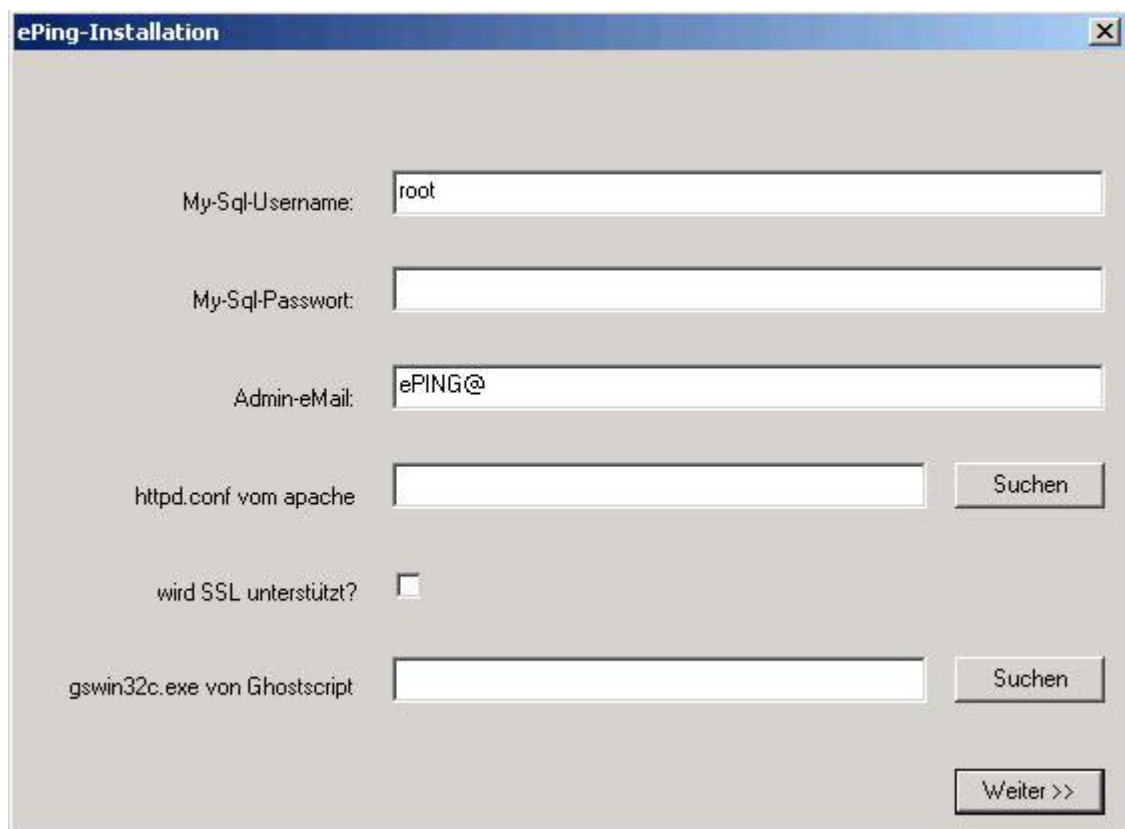


Abbildung 3: Eingabe aller notwendigen Parameter

- Editieren von php.ini
 - Setzen von register_global auf On
 - Eintragen bei prepend_files
 - `c:\eping\php\phplib-7.4\php\prepend.php` nach
(`c:\eping` = Installationsverzeichnis)
 - Bsp: `auto_prepend_file = "c:\eping\php\phplib-7.4\php\prepend.php"`
 - Eintragen im include_path
 - `c:\eping\exe;c:\eping\php` nach (mit `c:\eping` = Installationsverzeichnis)
 - Bsp: `include_path = ".;c:\eping\exe;c:\eping\php"`

Apache neu starten und ePing durch den Aufruf von `http://localhost/eping` testen.

Es wird automatisch ein Admin-User mit folgenden Parametern generiert:

Username: admin

Passwort: a_test1

1.5.2 Manuelle Anpassungen

Die folgende Anleitung ist nur dann relevant, wenn ePing nicht automatisch installiert wurde, oder Fehler auftraten.

Die Beschreibung gilt für das Installationsverzeichnis: `c:\eping` und den Zugriff auf ePing mittels `www.rechnerid.com/eping/`.

Zunächst muss die `php.ini` (sofern dies nicht schon erfolgt ist) von PHP angepasst werden. Hierzu ist die vom Apache genutzte `php.ini` maßgeblich. Meist ist diese unter `c:\windows` (`c:\winnt`) zu finden. Die genauen Anpassungen sind in den Kapiteln 1.4 und 1.5 beschrieben.

Sofern keine automatische Installation erfolgte, oder einzelne Komponenten verlegt wurden, sind alle Anpassungen, die die Lauffähigkeit von ePing gewährleisten, in der Datei `c:\eping\php\defines.php` vorzunehmen. Hier kann unter anderem

- User+Passwort für die MySQL-Datenbankzugriffe,
- der Pfad auf Ghostscript,
- das Installationsverzeichnis,
- Dokumentenverzeichnis (WWW-Zugriff auf ePing) und
- die eMail-Adresse (des Administrators)

angepasst werden.

Wurden die Datenbankinformationen nicht automatisch in MySQL eingeladen, kann dies durch das Kommando `source sva.mysql` direkt in MySQL nachgeholt werden. Die Datei `sva.mysql` findet man im Verzeichnis `c:\eping\install_scripts`.

Als letztes muss ePing noch für den Apache-Server sichtbar gemacht werden. Hierzu legt man entweder im Document-Root des Servers das Verzeichnis `eping` an und kopiert von `c:\ePing` das gesamte WWW-Verzeichnis hinein, oder man fügt in der Datei "httpd.conf" vom Apache (wird von der automatischen Installation genutzt) folgendes hinzu:

```
Alias /eping/ "c:/eping/www/"
Alias /eping "c:/eping/www/"
<Directory c:/eping/www/>
  Options -Indexes +FollowSymLinks
  AllowOverride None
```

```
Order allow,deny
Allow from all
DirectoryIndex index.php index.htm index.html index.phtml
</Directory>
```

Ist ein anderes WWW-Verzeichnis als eping gewünscht, kann dieses durch eine Anpassung sowohl in der defines.php, als auch in der httpd.conf erreicht werden.

Nach diesen Anpassungen, sollte ePing durch den Aufruf `http://localhost/eping` ansprechbar sein. Nun ist es nur noch notwendig, das im ePing-Dokumentenverzeichnis mitgelieferte phpAdmin gegen unbefugte Zugriffe abzusichern, da phpAdmin den Komplettzugriff auf die Datenbank erlaubt. Hierzu müssen folgende Zeile in die httpd.conf von Apache eingetragen werden:

```
<Directory "c:/eping/phpMyAdmin-2.2.6">
  AuthType Basic
  AuthName "Administrator"
  AuthUserFile "c:/eping/user/users"
  AuthGroupFile "c:/eping/user/groups"
  require group admin
</Directory>
```

1.5.3 Sicherheitsanpassung

1.5.3.1 Administrator von ePING

Hierzu loggt man sich als Administrator in ePing ein (User: admin, Passwort: a_test1). Man wählt hierzu den „Registrierten Bereich“ aus und geht in den „Adminbereich“. Dort folgt man dem Link zum User-Management. Durch das Eintragen eines neuen Usernamens und/oder Passwortes kann der Administrator angepasst werden. Dabei sollte nur darauf geachtet werden, dass die Rechte weiterhin dem ‚admin‘ zugeordnet sind. Das Passwort wird beim Speichern automatisch md5-Verschlüsselt und ist somit im Klartext zu schreiben.

Alternativ können diese Änderungen auch direkt in MySQL vorgenommen werden.

1.5.3.2 Administrator von phpMyAdmin

Im Verzeichnis „c:/eping/users“ die Datei „htpasswd.exe“ ausführen, um einen neuen User mit Passwort anzulegen.

```
htpasswd -c users LOGINNAME PASSWORT [erzeugt Userdatei neu]
htpasswd -u users LOGINNAME PASSWORT [neuen User einfügen oder updaten]
```

Danach muss dem User noch Gruppenrechte zugeordnet werden. Hierzu die Datei „groups“ editieren. Das Ergebnis könnte wie folgt aussehen:

```
admin:LOGINNAME
```

1.6 Deinstallation

Um ePing zu deinstallieren wählt man aus dem ePing-Programm-Verzeichnis "Uninstall ePing" und löscht danach das komplette Installationsverzeichnis. Zusätzlich sollten alle Anpassungen in den Dateien php.ini und mysql.conf per Hand rückgängig gemacht werden.

1.7 Typische Fehlermeldungen

Nach der Installation können folgende Fehler auftreten:

- Php-Script ‚xyz.php‘ könnte nicht gefunden werden
 - Kontrollieren Sie, ob in der php.ini ‚include_path‘ korrekt gesetzt ist
 - Starten Sie den Apache neu
- Beim Bestätigen einer Schaltfläche wie z.B. Berechnen, Einloggen passiert nichts
 - Kontrollieren Sie, ob in der php.ini register_globals = On eingestellt ist
 - Starten Sie den Apache neu
- Funktion ‚page_open‘ unbekannt
 - Kontrollieren Sie, ob in der php.ini prepend_files korrekt gesetzt ist
 - Starten Sie den Apache neu

2 Wartung

2.1 Lauffähigkeit des Systems

Das System ist ohne weitere Wartung lauffähig, doch sollten sicherheitsrelevante Updates, sowohl des Betriebssystems, als auch aller verwendeten Komponenten, vorgenommen werden. Weiterhin ist es zur Betreuung des Systems notwendig, dass der Apache-Server läuft.

Sollte der Server auf gar keine Anfrage mehr reagieren oder keine neuen Ergebnisse von den Rechenverfahren anzeigen, ist es am günstigsten den Rechner einfach neu zu starten, um die Funktionsfähigkeit wieder herzustellen.

2.2 Administrator

Der Administrator hat folgende Aufgaben im laufenden Betrieb des Systems:

- Usermanagement
 - Rechtevergabe
 - Es gibt folgende Usergruppen:
 - **admin** (volle Zugriffsrechte auf das Systems, außer PhpAdmin, welches einzeln erteilt werden muss)
 - **employee** (Zugriffsrechte auf den individuellen und Mitarbeiterbereich)
 - **individual** (Zugriffsrechte auf den Registrierten und den für diesen Benutzer freigegeben individuellen Bereich)
 - **user** (Zugriffsrechte auf den Registrierten Bereich)

- Anmerkung: Es können weitere Usergruppen eingerichtet werden. Siehe hierzu Kapitel 4.1
- Datenbankbetreuung über PhpAdmin
- Hat alle Rechte/Aufgaben die auch Mitarbeiter haben
- Neue Rechenverfahren einbinden bzw die Plattform betreuen und erweitern

2.3 Mitarbeiter

Die Mitarbeiter können neue individuelle Inhalte und Rechenverfahren einstellen, betreuen und die Zugriffsrechte für diese verändern. Dabei können nur Zugriffsrechte für die individuelle Nutzer vergeben werden, da Mitarbeiter immer volle Zugriffsrechte haben. Durch eine Erweiterung können einzelne Rechenverfahren auch nur für bestimmte Mitarbeitergruppen freigeschaltet werden - dies wird im Kapitel 4.2 beschrieben.

3 Aufbau der Dienstleistungsplattform ePING

3.1 Aufbau der internen Struktur von ePing (Verzeichnisse)

Grundsätzlich gliedert sich der interne Aufbau von ePing in 3 Bereiche:

- dem über einen Browser zugänglichen öffentlichen/eingeschränkten Bereich (siehe Kapitel 3.2)
- der Datenhaltung mittels MySql (siehe Kapitel 5)
- und den nicht Dateisystem (Skripte, Rechenverfahren etc.) Bereich auf der Festplatte

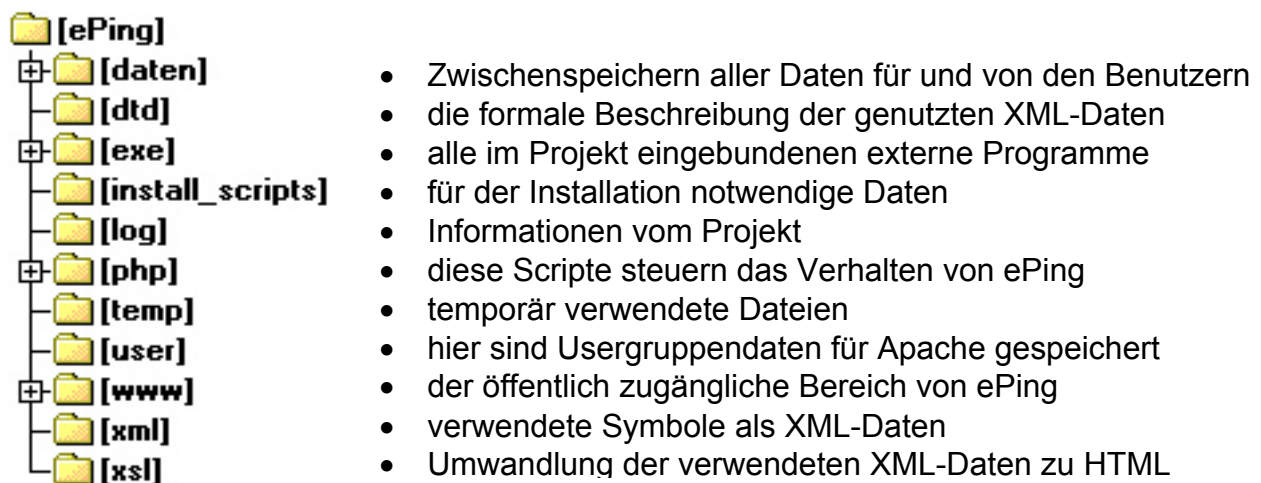


Abbildung 4: Dateistruktur von ePing

- **daten**

Dieses Verzeichnis dient als Schnittstelle zwischen den eingebundenen Rechenverfahren (welche auf der Basis von Eingabedateien und Ausgabedateien arbeitet) und ePing. Hierzu wird für jeden Benutzer ein eigenes Verzeichnis angelegt und in diesem die Daten gespeichert. Nach ca. 3 Wochen werden diese Dateien automatisch vom System entfernt.

- **dtd**

In diesem Verzeichnis liegen für alle Programme, die XML-Daten einlesen, die zugehörigen DTD's. Diese werden verwendet, um die hochgeladenen Eingabedaten auf syntaktische Korrektheit zu überprüfen. Erst wenn diese erfüllt ist, werden die Daten an das Rechenverfahren zur Bearbeitung weiter geleitet.

- **exe**

Hier liegen die ausführbaren Programme der einzelnen eingebundenen Anwendungen. Jedes dieser Rechenverfahren liegt in einem eigenen Verzeichnis und wird um spezifische PHP- und XML-Scripte ergänzt. Einzelheiten werden im Kapitel 4 beschrieben.

- **install_scripts**

Enthält den Datenbankaufbau, beschrieben durch ein MySql-Script, sowie das vom Installationsprogramm aufgerufene Startup-Programm, das notwendige Änderungen für die Lauffähigkeit von ePing vornimmt. Näheres wird im Kapitel 1.5 (automatische Installation), sowie im Unterkapitel 1.5.2 für manuelle Anpassungen beschrieben.

- **log**

In diesem Verzeichnis werden die Exec-Zugriffe von PHP geloggt, das heißt, es werden alle externen Zugriffe und die Art des Zugriffes auf diese Programme protokolliert. Zu finden sind diese Aufrufe in php/exec.php. Dies ist für eventuelle spätere Fehlersuche nützlich.

- **php**

Hier sind alle zentral genutzten PHP-Scripte von ePing gespeichert. Die genaue Funktionsweise wird im Kapitel 4 erläutert.

- **temp**

Dieses Verzeichnis nimmt nur kurzfristig Dateien auf, die in Zwischenschritte einer Berechnung erforderlich sind. Zum Beispiel bei der Umwandlung von Postscriptdateien in das gif-Format.

- **user**

Da ePing zur Administration der Datenbank phpMyAdmin nutzt, liegen in diesem Verzeichnis die vom Apache erforderlichen User und Gruppendaten, die den geschützten Zugriff auf phpMyAdmin gewährleisten. Weitere Erläuterungen hierzu sind im Kapitel 1.5.3.2 ausgeführt.

- **www**

Alle über einem Browser zugängliche HTML-Seiten und PHP-Scripte sind in diesem Verzeichnis untergebracht. Nähere Ausführungen über den Aufbau und die Funktionsweise sind in den Kapiteln 3.2 und 4 zu finden.

- **xml**

Zurzeit sind hier nur die von den Rechenverfahren genutzten Symbol und Einheitsdaten in einer XML-Datei gespeichert.

- **xsl**

Für jedes verwendete Rechenverfahren liegen in diesem Verzeichnis die XSL-Scripte, welche die Umwandlung der XML-Daten (Eingabe und Ausgabe der Programme) zu (im Benutzerbrowser darstellbare) HTML-Seiten übernimmt. Für die meisten Verfahren heißt dies, dass für jede unterstützte Sprache (zur Zeit Deutsch und Englisch) hier eine Eingabe-Xsl eine Ausgabe-Xsl-Datei abgelegt ist/erstellt wird.

3.2 Aufbau des öffentlichen und gesicherten Bereiches

3.2.1 Bereiche ePing, Conditions of Use, About ePing

In diesen Bereichen sind allgemeine Informationen zu ePing zu finden.

3.2.2 Bereich Free Area

Hier sind die öffentlich zugänglichen Rechenverfahren eingebunden. Näheres zum Einbinden von Rechenverfahren ab Kapitel 6.

3.2.3 Bereich Registered Area

Dieser Komplex beinhaltet alle für registrierte User zugängliche Rechenverfahren und weitere Informationen. In der Regel sind alle hier eingebundenen Inhalte für alle registrierten Kunden zugänglich, doch können entsprechende Einschränkungen getätigt werden. Zum Beispiel sind hier auch die nur den Mitarbeitern und Administratoren zugängliche Bereiche verlinkt. Diese Verlinkung ist aber nur mit entsprechenden Userrechten sichtbar. Hierdurch können auch Rechenverfahren eingebunden werden, welche nur von Mitarbeitern genutzt werden können.

Wie Rechenverfahren eingebunden und abgesichert werden, wird im Kapitel 6 beschrieben.

3.2.4 Bereich Individual Area

In diesem Bereich können weitere Rechenverfahren und speziell für einzelne Kunden aufgearbeitete Inhalte eingestellt werden, welche einzelnen oder allen registrierten Usern (mit dem

Zugriffsrecht auf den individuellen Bereich) zugänglich gemacht werden können. Näheres zum Einbinden von Rechenverfahren ab Kapitel 6.2.6 und zum Einstellen von anderen Inhalten im Kapitel 7.

3.2.5 Bereich Mitarbeiter

Die Mitarbeiter können alle Bereiche (außer den Administratorbereich) nutzen und haben für diese vollständige Rechte. Sie können somit für den Eigenbedarf jedes Rechenverfahren nutzen, aber auch Inhalte für bestimmte Kunden einstellen. Sollen bestimmte Rechenverfahren nicht allen Mitarbeitern zugänglich sein, kann dies durch die Erweiterung der Usergruppen erreicht werden. Näheres hierzu finden Sie im Kapitel 4.2.

3.2.6 Bereich Admin(istrator)

Der Administrator ist für das User-Management zuständig (Löschen und Anlegen von Usern, Zuordnen von Usern zu einer Usergruppe etc.). Weiterhin kann er die Zugriffe auf das System überwachen. Soll einem User mehr als eine Usergruppe zugeordnet werden, ist dies nur direkt über die Datenbank (phpMyAdmin) möglich. Hierzu müssen die Einzelnen Usergruppen mittels Komma getrennt werden.

4 Steuerung der Dienstleistungsplattform durch PHP

Die Plattform ePing wird vollständig durch PHP-Skripte gesteuert. In diesen Skripten werden die HTML-Seiten erzeugt. Doch können trotzdem reine HTML-Seiten in die Plattform eingebunden werden. Sollen diese Seiten aber nur bestimmten Usergruppen zugänglich sein, müssen sie durch PHP umschlossen werden. Dies kann man erreichen, indem die Datei zu <name>.phtml umbenannt wird und folgende Zeilen vor dem HTML-Code

```
<?  
page_open(array("sess" => "Example_Session", "auth" => "Example_Auth", "perm" => "Example_Perm"));  
?>
```

und folgende Zeilen am Ende des HTML-Codes

```
<?  
page_close();  
?>
```

hinzugefügt werden. Im Hintergrund wird diese Seite nun geschützt und nur registrierten Usern angezeigt.

Doch ist das Schützen von Seiten vor unbefugten Zugriffen nur eine Teilfunktionalität der hier zum Einsatz kommenden Werkzeuge. Wesentlicher für die Plattform sind die Funktionalitäten, welche das Einbinden der Rechenverfahren und das Arbeiten mit diesen ermöglicht. Hierzu sind alle wesentlichen genutzten Funktionen in Dateien im Verzeichnis /php zusammengefasst. Zum aktuellen Zeitpunkt gibt es folgende Dateien:

- **auto_include.php**
Diese Datei wird vor jedem Aufruf eines PHP-Skriptes automatisch eingebunden und included alle notwendigen Skripte, um die darin enthaltenen Funktionsdefinition sichtbar zu machen.
- **bild_bearbeitung.php**
Hier sind alle Funktionen zusammengefasst, welche die Umwandlung von Bildern eines Formates (PS) in ein anderes Format (JPG,GIF,BMP) notwendig sind, sowie Funktionalitäten, welche das Ausschneiden von Teilbilder ermöglicht.
- **customer_db.php**
Die darin enthaltene Klasse, ermöglicht es die Userdaten zu manipulieren und abzufragen.
- **db.php**
Diese Klasse dient nur dazu, um den Zugriff auf die Datenbank zu gewährleisten. Dies wird durch das Setzen des Datenbanknamens, des Usernamens und des Userpasswortes erreicht.
- **defines.php**
Hier werden alle globalen Variablen und Abhängigkeiten verwaltet. Wird das Projekt bewegt, oder der WWW-Zugriff ändert sich, muss dies in dieser Datei angepasst werden.
- **documentation_html.php**
Gegenwärtig stehen zwei Funktionen zur Verfügung, die einen Standardkopf und einen Standardfuß für die Anzeige einer Dokumentation erzeugen.
- **error_table_html.php**
Damit die Anzeige von fehlerhaften Eingaben einfach ausgegeben wird, sind hier standardisierte Funktionsaufrufe zusammengefasst.
- **exec.php**
Diese Klasse ermöglicht es Zugriffe auf externe Programme (Rechenverfahren) vorzunehmen. Das Ganze wird über einen Systemaufruf des Programms realisiert.
- **files.php**
Die Funktionen helfen beim Arbeiten mit dem Dateisystem. Es kann überprüft werden, ob ein bestimmtes Verzeichnis existiert oder es können damit Verzeichnisse angelegt werden. Außerdem werden von einer entsprechenden Funktion alte Dateien/Daten aus den Userverzeichnissen (nach 3 Wochen) gelöscht.
- **functions.php**
In dieser Datei sind alle weiteren häufig verwendeten Funktionen zusammengefasst. Dies reicht vom Ausgeben der Administrator-Email-Adresse bis zum Test ob eine Eingabe einen bestimmten Wertebereich erfüllt.
- **html.php**
Hier sind alle das Aussehen steuernden Funktionen enthalten.
- **info.php**
Durch das Verwenden der hier hinterlegten Funktionalitäten kann eine kleine Statistik über die Zugriffe auf das System erstellt werden.

- **log.php**
Hier werden Seiten- und Programmaufrufe geloggt.
- **page_db.php**
Diese Klasse steuert den Zugriff und das Einstellen der gesicherten Inhalte im individuellen Bereich der Plattform.
- **programm.php**
Hier wird die Steuerung, sowie die Eingabe und Ausgabe für jedes eingebundene Rechenverfahren vorgenommen.
- **secure.php**
Durch entsprechenden Test entscheiden die zwei mitgelieferten Funktionen, ob ein User Zugriff auf das Rechenverfahren oder den Inhalt im individuellen-Bereich erhält oder nicht.
- **usergroup.php**
In dieser Datei können neue Usergruppen eingerichtet werden (Näheres im nächsten Kapitel)
- **xml_parse.php**
Durch die hier ausgelagerten Funktionalitäten können aus einer korrekt erstellten XML-Datei die enthaltenen Daten als Variablen extrahiert werden.

4.1 Usergruppen einrichten

Unter Umständen reichen die 4 vordefinierten Usergruppen nicht aus. So kann es notwendig sein, nur bestimmten Mitarbeitergruppen einzelne Rechenverfahren zuzuordnen. Um nicht jedes Rechenverfahren für jeden User einzeln freizuschalten, kann für eine Gruppe von Usern eine neue Usergruppe angelegt werden. Hierzu muss die Datei `.../php/usergroup.php` angepasst werden. Die Usergruppen und ihr Wert sind in einem assoziativen Array gespeichert.

```
$_eping_permissions_outside =  
array(  
    "user" => 1,           // Bit: 00000001  
    "individual" => 3,     // Bit: 00000011  
    "employee" => 7,      // Bit: 00000111  
    "admin" => 255        // Bit: 11111111  
);
```

Hierbei wird dem Namen der Usergruppe ein binärer Wert zugeordnet. So ist es möglich hierarchische Abstufungen einzubauen. In der Regel sollte die Admin-User-Gruppe alle Rechte bekommen, und somit alle gesetzten Bits jeder anderer Usergruppen auch gesetzt bekommen. Bei allen anderen Usergruppen kann eine hierarchische oder auch sich gegenseitig ausschließende Wertzuweisung erfolgen. So wird zum Beispiel durch das Einführen folgender zwei Gruppen:

```
"gruppe_1" => 15         //Bit: 0001111  
"gruppe_2" => 23         //Bit: 0010111
```

zwei Usergruppen definiert, welche gegenseitig nicht auf explizit für die Usergruppe eingerichtete Bereiche zugreifen dürfen. Doch können beide Gruppen weiterhin auf freigegebene Inhalte für die Usergruppen "user" und "individual" zugreifen.

Anmerkung: Die Zuordnung von Usern zu Usergruppen kann der Administrator im Bereich 'Usermanagement' vornehmen.

4.2 Rechenverfahren einzelnen Usergruppen zuordnen

Um Rechenverfahren einzelnen Usergruppen zuzuordnen, müssen diese außerhalb des individuellen Bereiches eingebunden werden. (Im individuellen Bereich würde dies zwar auch funktionieren, doch würde man dort die Rechenverfahren unter Umständen sehen, auch wenn man sie dann nicht nutzen kann)

Die allgemeine Vorgehensweise entspricht dem Einbinden eines Verfahrens im registrierten Bereich. (siehe Kapitel 6 ff.) Nachdem dies geschehen ist, muss das PHP-Skript `.../www/<name>/<name>.php` angepasst werden. (siehe Kapitel 6.2.3.1) Zunächst würde das Skript ungefähr wie folgt aussehen:

```
<?php

page_open(array("sess" => "Example_Session", "auth" => "Example_Auth", "perm" => "Example_Perm"));

include("holtrop/holtrop_main.php");
include("holtrop/holtrop_error_table.phtml");

$sis_bild=1;
set_variablen("holtrop", "english", "upload", $sis_bild);
adjust_variable_if_load_old("holtrop", $sis_bild);

// HAUPTTEIL
programm_kopf();
programm_handle_upload($language, $userfile, $uploaded_file, $command);
programm_aufruf();
programm_enable_download($language, $dateibild, $dateips, $dateixml, $dateixml_out);
programm_fuss();

page_close();

?>
```

Um das Rechenverfahren nur einer bestimmten Usergruppe (und hierarchisch höher gestufte Usergruppen) zugänglich zu machen, muss nach dem `page_open` Funktionsaufruf folgender Aufruf erfolgen:

```
if (!$perm->have_perm("user_gruppen_name"))
{
    my_error("Permission denied.");
    ... //tue noch ein paar andere Sachen
    page_close();
    exit;
}
```

Hat ein User nicht die Rechte dieser Gruppen, erfolgt die Ausgabe einer Fehlermeldung und das Beenden des Skriptes. Durch eine leichte Anpassung dieser Befehlszeilen, kann so auch

bestimmten Usergruppen etwas anderes, bzw. weitere Informationen ausgegeben werden. Zum Beispiel in folgender Form:

```
if ($perm->have_perm("user_gruppen_name"))  
{  
    ... //zeige Link für Rechenverfahren  
}
```

5 Datenhaltung

Die Datenhaltung von ePing erfolgt in einer MySQL Datenbank. Durch leichte Änderungen im PHP-Script ist auch das Einbinden anderer Datenbanken möglich. Die Datenbank ist in vier große Bereiche untergliedert:

- User- und Session-Daten
- Programm-Daten (Eingabe + Ausgabe der Rechenverfahren)
- Rechte-Management-Daten
- Log-Daten

Der genaue Aufbau der Datenbanktabellen ist der Datei `install_scripts/sva.mysql` zu entnehmen.

Anmerkung: Für eine laufende Session, erfolgt die Persistenzhaltung (sofern aktiv) auch in der Datenbank, die Daten werden jedoch direkt aus den erzeugten Dateien gelesen und ausgegeben.

6 Anleitung zur Integration von Rechenverfahren

6.1 Anforderung an die Rechenverfahren

- Das einzubindende Rechenverfahren muss in ausführbarer Form vorliegen.
- Die Rechenverfahren müssen immer terminieren und nicht zum Beispiel fehlende Daten per Kommandozeile selbständig anfordern. Kann die Terminierung für bestimmte Eingaben nicht gewährleistet werden, müssen dies vorher abgefangen werden.
- Da das Rechenverfahren extern angesprochen wird, hat ePing keinen Einfluss darauf, wie viel Rechenzeit dieses benötigt. Sollte eine Rechnung sehr lange dauern, sind zusätzliche Anpassungen erforderlich. Nähere Informationen finden Sie hierzu im Kapitel 6.2.8.
- Die notwendigen Eingabe- und Ausgabedaten werden dem Rechenverfahren durch Dateien übermittelt. Bei den Eingabedaten kann das System auch so angepasst werden, dass es eine Eingabepipeline unterstützt.
- Die Eingabe- und Ausgabedateien müssen variable Dateinamen unterstützen (erforderlich für parallele Abarbeitung).
- Als Parameter für die Eingabedateien sollten komplette Zuordnungen (Pfad+Dateinamen) unterstützt werden. (Ansonsten müssen weitere Anpassungen vorgenommen werden. Dies trifft insbesondere dann zu, wenn die Eingabe- und Ausgabedateien nur im selben Verzeichnis wie das ausführbare Rechenverfahren liegen dürfen. Speziell sollten längere Dateinamen als die 8-Zeichen langen DOS-Dateinamen unterstützt werden).
- Intern erfolgt der Programmaufruf standardmäßig wie folgt:
<name> <input_datei> <output_datei> <output_bild> <output_log>
Hierbei sind

- <output_bild> eine optionale Ausgabe, zum Beispiel einer PS-Datei
 - <output_log> eine optionale Ausgabe, zum Beispiel Erfolgs/Fehlermeldung
- Der Programmaufruf kann für jedes Rechenverfahren überschrieben und angepasst werden. Siehe hierzu das Kapitel 6.2.3.5.

Die Beschreibung gilt für ein Rechenverfahren, das in den registrierten Bereich eingestellt werden soll. Die Anpassungen, welche notwendig sind, um ein solches Verfahren im individuellen Bereich oder öffentlichen Bereich einzubinden, werden später erläutert.

6.1.1 Notwendige Dateien

Das Rechenverfahren bekommt einen eindeutigen zusammengeschrriebenen Namen – zum Beispiel ‚holtrop‘. Durch diesen Bezeichner werden die Namen der im folgenden aufgelisteten notwendigen Dateien vorgegeben. Optional sind hierbei alle Dateien mit Sprachkomponenten, die nur erstellt werden müssen, wenn diese Sprache unterstützt werden soll. Ohne Angabe wird die Datei als eine englischsprachige Datei angenommen. Das Projekt wird zweisprachig ausgeliefert und unterstützt neben Englisch auch Deutsch. Dabei wird bei deutschen Scripten ‚german‘ verwendet. Steht für bestimmte Rechenverfahren keine entsprechendes Modul zur Verfügung, sollte das Default-Modul eingelinkt werden.

Liste der notwendigen Dateien mit der korrekten Dateinamensyntax:

<name>.dtd	notwendig, um Eingabe XML-Daten zu validieren.
<name>.exe	das eigentliche Rechenverfahren
<name>_first.xml	Defaultdaten, welche beim Start der Eingabemaske angezeigt werden sollen
<name>_reset.xml	Defaultdaten, welche beim Drücken der Reset-Submit-Schaltfläche angezeigt werden sollen
<name>.php	das zur Steuerung der Eingabe/Berechnung/Ausgabe notwendige Script
<name>_<sprache>.php [optional]	eine weitere unterstützte Sprache für ein Rechenverfahren
documentation\<name>.phtml	enthält die Dokumentation des Rechenverfahren
<name>_main.php	liefert die für jedes Rechenverfahren notwendige Funktionalitäten, welche immer dieselben Funktionsrumpfe besitzen. z.Bsp: die Fehlererkennung <function error_check()> Funktionalitäten können auch in andere Dateien ausgelagert werden, doch müssen diese dann dem Hauptsript zugänglich gemacht werden. In der Regel wird die Fehlerbeschreibung ausgelagert, um eine Mehrsprachigkeit leichter zu unterstützen. Bsp: <name>_error_table.phtml <name>_error_table_german.phtml
<name>.xsl	wandelt die Eingabe XML-Daten zu einer HTML-Ansicht um
<name>_out.xsl	wandelt die Ausgabe XML-Daten zu einer HTML-Ansicht um
<name>_<sprache>.xsl [optional]	wandelt die Eingabe XML-Daten zu einer HTML-Ansicht der Sprache <sprache> um.
<name>_<sprache>_out.xsl [optional]	wandelt die Eingabe XML-Daten zu einer HTML-Ansicht der Sprache <sprache> um.

Erfolgt die Ausgabe als reine Textdatei, entfallen alle Komponenten der Form *_out*.

6.1.2 Arbeitsschritte zum Einbinden eines Rechenverfahren

1. Es müssen folgende Unterverzeichnisse angelegt werden:
 - exe*<name>*
 - www\register*<name>*
2. Das ausführbare Programm wird nach exe*<name>**<name>*.exe kopiert.
3. Im Verzeichnis exe*<name>* die Default-Xml-Dateien
 - *<name>*_reset.xml
 - *<name>*_first.xml
 - *<name>*_main.php
 - *<name>*_error_table.phtml
 - *<name>*_error_table_*<sprache>*.phtmlerstellen.
4. Im Verzeichnis dtd die Default-Dtd-Datei
 - *<name>*.dtderstellen
5. Im Verzeichnis xsl die Default-Xsl-Dateien
 - *<name>*.xsl
 - *<name>*_out.xsl
 - *<name>*_ *<sprache>*.xsl
 - *<name>*_out_*<sprache>*.xslerstellen.
6. Für den späteren Download der Eingabe/Ausgabedaten des Rechenverfahrens ist das Rechenverfahren in der Datei „registered/download.php“ bekannt zu geben.
7. Das Rechenverfahren in die Oberfläche verlinken.

6.2 Einbinden eines externen Programms (XML)

Nachfolgend wird beispielhaft an dem Programm „Holtrop“ beschrieben, wie ein Rechenverfahren in die Plattform eingebunden werden kann.

6.2.1 Vorarbeiten

1. Erstellen der notwendigen Verzeichnisstruktur

exe\holtrop
www\register\holtrop

2. Kopieren der ausführbaren Datei (gegebenenfalls umbenennen der ausführbaren Datei)

holtrop.exe nach *exe\holtrop\holtrop.exe*

3. Erstellen der Default-XML-Datei

holtrop_first.xml und holtrop_reset.xml in *exe\holtrop*

In der *_first.xml sind die Daten einzutragen, welche beim Erstaufwurf des Rechenverfahrens angezeigt werden sollen. Dabei muss die vollständige XML-Struktur aufgeführt werden, da nur so alle Eingabefelder vorhanden sind. Im Zweifel bleibt der Inhalt eines Tags leer. In der *_reset.xml sollten die Daten aufgeführt werden, welche beim Drücken der Reset-Schaltfläche erscheinen sollen. In der Regel wird dies der leere Datensatz sein. Auch hier gilt, dass die komplette XML-Struktur angegeben werden muss. Um sicherzustellen, dass die XML-Struktur korrekt ist, kann diese mit der zugehörigen DTD validiert werden.

4. Erstellen der zugehörigen DTD für die Eingabe XML-Struktur

dtd\holtrop.dtd

5. Erstellen der XSL-Dateien

Siehe Kapitel 6.2.2

6. Erschaffen der Funktionalität für das Rechenverfahrens

Siehe Kapitel 6.2.3

7. Erstellen der Dokumentation

Die Dokumentation ist ein normale "HTML-Datei", welches per PHP um Elemente für die Navigation erweitert wird (Daraus resultiert eine phtml-Datei). Die Dokumentation wird unter `www\documentation\ abgelegt.`

Sollte keine Dokumentation vorliegen, kann das Einbinden einer Dokumentation per Flag unterbunden werden. Hierzu muss im Hauptscript das Flag `$no_documentation=1;` gesetzt werden.

6.2.2 Erstellen der XSL-Datei

6.2.2.1 Anforderungen an die XML-Struktur

Um die Zuordnung der im Formular erfassten Daten zu der XML-Struktur zu gewährleisten, sollte jeder verwendete Tag einzigartig sein. Intern wird im Formular immer der Tag als Name für das HTML-Eingabefeld verwendet. Würde dasselbe Tag an unterschiedlichen Stellen in der XML-Datei verwendet werden, würde dies zum Überschreiben der Daten beim Senden des Formulars führen. Das könnte man umgehen, indem im Formular für solche Elemente andere Bezeichner verwendet werden, doch stellt dies immer eine mögliche Fehlerquelle dar. Deshalb sollte schon beim Aufbau der XML-Struktur auf Eindeutigkeit geachtet werden. Das ganze gilt nicht für Wiederholungen derselben Struktur innerhalb der XML-Daten.

Beispiel XML Teilstruktur:

```
<Sections>
  <S>1</S>
</Sections>
<Sections>
  <S>2</S>
</Sections>
```

DTD:

```
<!ELEMENT Sections ( S )>  
<!ELEMENT S (#PCDATA)>
```

ist erlaubt, aber

```
<Sections>  
<S>  
  <X>1</X><S>.1</S>  
</S>  
</Sections>
```

DTD:

```
<!ELEMENT Sections ( S )>  
<!ELEMENT S ( (X, S) | #PCDATA)>
```

ist nicht erlaubt, da hier unterschiedliche Objekte mit demselben Tag bezeichnet werden. Am leichtesten ist diese Richtlinie in der zugehörigen DTD zu überprüfen.

6.2.2.2 Die XSL-Datei

Sollen die Eingabe-/Ausgabedaten dargestellt werden, so müssen die XML-Daten durch eine XSL-Transformation in HTML umgewandelt werden. Hierzu werden die XML- und die XSL-Datei zusammen mit freien Parametern intern verknüpft. Diese freien Parameter müssen in jeder XSL-Datei verarbeitet werden. Hierdurch und durch das vorgegebene Layout ergibt sich für jedes Programm ein bestimmter gleich bleibender Aufbau der XSL-Datei, welcher dann um die spezifischen Elemente das Programm erweitert wird.

Beachte: die XSL-Transformation soll keine komplette HTML-Datei erzeugen, da das Ergebnis noch nachträglich um Navigation und andere Elemente erweitert wird. Hier soll nur das Layout für die Eingabe-/Ausgabeelemente erzeugt werden.

Der Grundaufbau sieht wie folgt aus:

```
<?xml version="1.0"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
<xsl:output method="html" standalone="no" indent="yes"/>  
<xsl:param name="random_name"/>  
<xsl:param name="programm_name"/>  
<xsl:param name="aip"/>  
  
<xsl:template match="/">  
  <FORM method="post"><xsl:attribute name="action"><xsl:value-of  
select="$programm_name"/>.php </xsl:attribute>  
  <INPUT type="hidden" name="random_name">  
    <xsl:attribute name="value"><xsl:value-of select="$random_name"/></xsl:attribute></INPUT>  
  <INPUT type="hidden" name="aip">  
    <xsl:attribute name="value"><xsl:value-of select="$aip"/></xsl:attribute></INPUT>  
<TABLE bgcolor="#ffffff" border="0" cellpadding="0" width="100%">  
<TR>  
<TD>  
  <xsl:apply-templates select="[*ROOT_DATEN_TAG]"/>
```



```
</TD>
</TR>
</TABLE>
</FORM>
</xsl:template>

<xsl:template match="[*ROOT_XML_TAG*]">
[*Konkretes Layout*]
</xsl:template>

</xsl:stylesheet>
```

Hierbei beschreiben die ersten 3 Zeilen die Art des Dokumentes. Die nächsten 3 Zeilen beschreiben die drei freien Parameter, welche übergeben würden. Diese Elemente sind zur Navigation zwischen der Eingabeseite (den Eingabeseiten) und der Ausgabeseite notwendig.

Im folgenden wird nun die XML-Datei geparsed und entsprechend der gefundenen Matches die Ausgabe erzeugt. Zunächst wird das Root-Tag gefunden und ein Table-Element eingebettet in ein Form-Element erzeugt. In diesem wird explizit der Anfang der XML-Daten aufgerufen. Hierzu muss [*ROOT_DATEN_TAG*] durch den entsprechenden Tag-Namen in der XML-Datei ersetzt werden. Dies wird in der Regel der Programmname sein. Bsp: `<xsl:apply-templates select="holtrop"/>` Dieselbe Änderung nimmt man bei der zugehörigen Match-Sektion vor. Bsp: `<xsl:template match="holtrop">` Danach wird das konkrete Layout innerhalb dieser Sektion erstellt. Dabei wird es in der Regel notwendig sein, bestimmte Aktionselemente einzufügen. Standardmäßig verwendet wurden folgende Schalterelemente:

Berechnung: Die Eingabe wird abgeschlossen und das Ergebnis erzeugt

```
<INPUT class="bu" type="submit" name="calc" value="Calculate"/>
```

Weiter: Es folgt eine weitere Eingabeseite

```
<INPUT class="bu" type="submit" name="next_page" value="Next"/>
```

Reset: Die Daten in den Feldern werden gelöscht

```
<INPUT class="bu" type="reset" name="reset" value="Reset"/>
```

bzw – wenn der Inhalt geleert werden soll:

```
<INPUT class="bu" type="submit" name="reset" value="Reset"/>
```

Diese Elemente sind in der Regel in einer Tabelle eingelagert. Als Beispiel folgen nun eine Weiter und eine Reset-Schaltfläche:

```
<TABLE border="0" cellPadding="0" cellSpacing="1" width="100%">
<TR>
  <TD align="right"><INPUT class="bu" type="submit" name="next_page" value="Next"/></TD>
  <TD align="left"><INPUT class="bu" type="submit" name="reset" value="Reset"/></TD>
</TR>
</TABLE>
```

6.2.2.3 Ausgabe XSL-Datei

Die Erstellung der Ausgabe-XSL-Datei entspricht weitgehend der Erstellung der Eingabe-XSL-Datei, nur dass hier keine zusätzlichen Schaltflächen notwendig sind und das Layout ohne Eingabefelder auskommt.

6.2.2.4 Ausgabe als reine Textdatei einbinden (Non-XML-Ausgabe)

Erfolgt die Ausgabe als reine Textdatei, kann diese direkt angezeigt werden, ohne zuvor in XML umgewandelt werden. Hierzu gibt es ein Flag welches dieses Verhalten regelt. Bsp:

```
$is_bild=1;  
$is_text_ausgabe=1;  
set_variablen("holtrop", "english", "upload", $is_bild, $is_text_ausgabe);
```

6.2.2.5 Andere XSL-Sprach-Dateien

Hierfür werden die Original-Xsl-Dateien mit dem dazugehörigen Sprachzusatz kopiert und danach intern entsprechend angepasst.

6.2.2.6 Anpassung der XSL bei Aufteilung der Eingabedaten über mehrere Seiten

Sollen die Eingabedaten über mehr als eine Seite verteilt werden, ist zunächst eine einzige XSL-Datei zu erzeugen. Diese wird dann entsprechend der Aufteilung kopiert. Danach wird jede Datei so angepasst, dass alle gerade nicht benötigte Elemente nur noch versteckt enthalten sind. Hierzu sind die zugehörigen Layoutelemente zu entfernen und das Eingabefeld wird durch das Setzen auf den Typ=hidden versteckt. Bsp:

```
<INPUT type="hidden" name="RHO">  
  <xsl:attribute name="value"><xsl:value-of select="RHO"/></xsl:attribute>  
</INPUT>  
<INPUT type="hidden" name="NY">  
  <xsl:attribute name="value"><xsl:value-of select="NY"/></xsl:attribute>  
</INPUT>
```

Zum Schluss müssen noch die Schaltflächen angepasst werden, so dass alle Eingabeseite bis auf die letzte eine 'Next'-Schaltfläche und die letzte eine 'Calculate'-Schaltfläche besitzen.

6.2.3 Erstellen der PHP - Funktionalität des Rechenverfahren

Um ein Rechenverfahren einzubinden, müssen folgende PHP-Skripte erstellt werden:

Im Verzeichnis exe\<<name>

- <name>_main.php
- <name>_error_table.phtml

Im Verzeichnis www\register\<<name>

- <name>.php

und ggf. um zusätzliche Spracherweiterungen ergänzt werden. Nachfolgend wird wieder davon ausgegangen, dass das Rechenverfahren ‚Holtrop‘ eingebunden werden soll.

6.2.3.1 Holtrop.php

Dieses Skript dient als Schnittstelle zwischen dem Benutzer an seinem Browser und der Plattform. Damit das Rechenverfahren für einen Benutzer sichtbar ist, muss dieses in die Web-Oberfläche eingebunden (verlinkt) werden. In der Regel wird dies in der Datei ,www\register\methods.php' erfolgen, doch ist dies auch wo anders möglich, genauso wie obige Datei ,Holtrop.php' nicht zwingend in dem Verzeichnis www\register\holtrop liegen muss.

Im weiteren soll der Aufbau der Datei beschrieben werden, der fast immer gleich ist.

```
<?php
page_open(array("sess" => "Example_Session", "auth" => "Example_Auth", "perm" => "Example_Perm"));

include("holtrop/holtrop_main.php");
include("holtrop/holtrop_error_table.phtml");

$sis_bild=1;
set_variablen("holtrop", "english", "upload", $sis_bild);
adjust_variable_if_load_old("holtrop", $sis_bild);

// HAUPTTEIL
programm_kopf();
programm_handle_upload($language, $userfile, $uploaded_file, $command);
programm_aufruf();
programm_enable_download($language, $dateibild, $dateips, $dateixml, $dateixml_out);
programm_fuss();

page_close();

?>
```

Hierbei wird das Skript von den zwei Befehlen ,page_open' und ,page_close' umschlossen, welche sicherstellen, dass diese Seite nur von jemandem eingesehen werden kann, der auch im System eingeloggt ist. Durch eine einfache Erweiterung könnte man den Zugriff noch zusätzlich weiter einschränken. Nähere Informationen finden Sie im Kapitel 4.2. Danach werden die zwei noch zu erstellenden Skripte ,holtrop_main.php' und ,holtrop_error_table.phtml' eingebunden, um die Funktionalität dieser sichtbar und benutzbar zu machen. Auffallend ist hierbei, dass Teile des Pfades beim Einbinden mit angegeben werden müssen. Dies liegt daran, dass im ,include_path' von php.ini nur das Hauptverzeichnis ,exe' eingebunden wurde und nicht für jedes Rechenverfahren auch noch das Unterverzeichnis.

Die folgenden Befehle steuern nun die Funktionalitäten des Rechenverfahrens und legen ein bestimmtes Verhalten fest und sollen nun näher betrachtet werden. Die Funktion ,set_variablen' hat folgende Funktionsrumpf:

```
set_variablen($method, $language, $uploadname, $sis_pic=0, $sis_text=0)
```

- \$method erwartet den Namen der Methode. Dies wird genutzt, um die Pfade korrekt zu bestimmen. In unserem Fall müssen wir somit ,holtrop' angeben. (Intern wird daraus unter anderem ,holtrop.exe' gebildet, um das korrekte Rechenverfahren ansprechen zu können)
- \$language erwarten den Namen des zu nutzenden Sprachpaketes. Ist dieser Wert leer (") oder kann das entsprechende Sprachpaket nicht gefunden werden, wird automatisch auf kein Sprachpaket umgeschaltet. (Intern wird daraus unter anderem ,holtrop_german.xml' gebildet, wenn hier ,german' übergeben wurde, um so die deutsche Eingabeseite einzubinden.)

- `$uploadname` ist obsolete und kann mit irgend etwas gefüllt werden
- `$is_pic` ist ein optionaler Parameter. Wird dieser gesetzt, versucht das System zusätzlich eine graphische Ausgabe einzubinden. Hierbei wird davon ausgegangen, dass diese Ausgabe als PostScript-Datei vorliegt und zunächst nach JPG umgewandelt werden muss.
- `$is_text` ist ein optionaler Parameter. Ist dieser gesetzt, führt dies dazu, dass die Ausgabe des Rechenverfahrens direkt ausgegeben wird. Dies sollte immer dann gemacht werden, wenn keine XML-Datei erzeugt wird.

Der nächste Funktionsaufruf (*`adjust_variable_if_load_old($method,$is_bild)`*) dient dazu, um intern Anpassungen vorzunehmen, falls eine Eingabe(xml)datei hochgeladen wurde. Die zwei zu übergebenen Parameter sind genau dieselben wie im Funktionsaufruf darüber.

Die folgenden fünf Funktionsaufrufe müssen jeweils einfach nur übernommen werden. Sie ändern sich nie. Sie dienen dazu, um die Anzeigeseite (unterschiedlich) aufzubauen und die Rechnung zu steuern.

Außer den bis jetzt erklärten Funktionen und Parameter gibt es noch weitere, die das Verhalten steuern können. Die wichtigsten sind

- *`set_anzahl_input_pages(6);`*
Dieser Funktionsaufruf muss direkt nach dem `page_open` erfolgen und sagt dem System, dass mehr als eine Eingabeseite vorhanden ist.
- *`$bild_x=120; $bild_y=120; $bild_breite=605; $bild_hoehe=830;`*
Das Setzen dieser Variablen führt dazu, dass aus der erzeugten PS-Datei ein anderer Ausschnitt ausgeschnitten und als JPG-Bild umgewandelt wird. In diesem Fall wird ein doppelt so hohes Bild erzeugt, als standardmäßig eingestellt ist.

6.2.3.2 Holtrop_main.php

In dieser Datei werden die wichtigsten zu einem Rechenverfahren gehörenden Funktionen zusammengefasst. Hier wird alles bis auf die Ausgabe der Fehlermeldungen erledigt. Hierzu gehört das Erzeugen der Eingabe-XML Datei, das Überprüfen des Wertebereiches von Eingaben und das Sichern der Eingabe zwischen einer möglichen Fehlerseite und der vorherigen Eingabeseite. Weitere Rechenverfahren spezifische Funktionalitäten werden auch hier erzeugt, doch sind in der Regel nur folgenden Funktionen definiert:

```
function datei_testen($xml_file = "daten.xml",$dtd_file = "daten.dtd")
function datei_schreiben($xml_file = "daten.xml", $mehr_daten = 0, $error_code )
function save_inputs()
function error_check()
```

Die erste Funktion kann dazu genutzt werden, um die erzeugte XML-Eingabedatei vor der Übergabe an das Rechenverfahren zu überprüfen. Da jedoch die XML-Datei selbst erzeugt wird, ist dieser Funktionsaufruf einfach leer. (Anmerkung: Beim Hochladen von XML-Eingabe_Dateien wird das Überprüfen der syntaktischen Korrektheit an anderer Stelle übernommen.)

Die nächste Funktion – `datei_schreiben` – übernimmt das Erzeugen der XML-Datei. Hierzu müssen alle Eingabevariablen global gesetzt werden, um auf den Inhalt dieser zugreifen zu können. Welche Variablen alles benötigt werden, ist aus den Benennungen der Eingabefelder der Eingabeseite ersichtlich. Über einen möglichen Aufbau kann man bei den mitgelieferten Rechenverfahren Aufschluss erlangen.

Die Funktion `save_inputs` dient dazu, die Eingabewerte vor einer Fehlerkorrektur zwischen zu speichern. Hierzu ruft man innerhalb der Funktion `return save_all_vars` auf und übergibt an diesen Funktionsaufruf ein Array von den genutzten VariablenNAMEN. Es ist zu beachten, dass hier nicht die Variablen, sondern die Variablennamen übergeben werden!

Zum Schluss folgt noch die Funktion `error_check()`. Hier kann man die Überprüfung der Eingabe auf korrekten Datentyp und Wertebereich vornehmen. Näheres erschließt sich aus den mitgelieferten Rechenverfahren.

6.2.3.3 Holtrop_error_table.phtml

In dieser Datei gibt es nur eine Funktion. Sie ist dafür zuständig, dass einem Fehlercode eine Beschreibung zugeordnet wird. Deshalb ist diese Funktionalität auch ausgelagert, um mehr als eine Sprache zu unterstützen. Soll der Fehlercode in einer anderen Sprache angezeigt werden, muss nur die entsprechende Datei included werden.

Der Aufbau ist immer gleich. Es wird die Funktion:

```
function show_error_table($error_code, $save_inputs, $url)
```

angesprochen. Als Übergabe erhält diese Funktion den Fehlercode, die zwischengespeicherte Eingabe und die Rücksprung-URL. Im ersten Teil des Skriptes werden Variablen gesetzt, die direkt ausgegeben werden. Für andere Sprachen müssen somit diese Inhalte angepasst werden. Für deutsch würde das ganze wie folgt aussehen:

```
$pre_back = "Ihre Eingabedaten sind nicht korrekt. Bitte";  
$back_name = "&#x000D";  
$post_back = "Sie Ihre Eingaben.";  
$name = "Flachwassereinfluß auf die Schiffsgeschwindigkeit (Fehler)";  
$tab1 = "Beschreibung";  
$tab2 = "Parameter";  
$tab3 = "Eingabe";  
$tab4 = "Grenzwerte";  
$color= 1;
```

Anmerkung: `$color` beschreibt mit welcher der beiden vordefinierten Farben die Tabellenzeilen auf der Fehlerseite anfangen. Anschließend wird dann die Fehlerseite aufgebaut. Sie hat immer die Form:

```
error_kopf($url,$pre_back,$back_name,$post_back,$name,$tab1,$tab2,$tab3,$tab4);  
if ($error_code&1) show_error($color,"Mittlerer Tiefgang","TM",$save_inputs["TM"],"0.1 ... 20.0");  
... <weiter Fehlermeldungen> ...  
error_fuss();
```

`error_kopf` und `error_fuss` starten bzw. beenden die Fehlerseite. Dazwischen wird bei entsprechendem Fehlerauftreten immer eine Zeile mit der Beschreibung des entsprechenden Fehlers ausgegeben. Für weitere Informationen sehen sie bitte bei den mitgelieferten Rechenverfahren nach.

6.2.3.4 Bilder anzeigen

Bis jetzt, wurde nur dargelegt, wie Ausgabegrafiken angezeigt werden können, die vom Rechenverfahren automatisch erzeugt wurden. Dabei geht das System davon aus, dass die

entsprechende Grafik zunächst in einer PostScript-Form vorliegt und intern nach JPG umgewandelt werden muss. Das wird durch den Funktionsaufruf *set_variablen* gesteuert. Dabei wird dem Rechenverfahren ein weiterer Parameter übergeben, damit dem Verfahren bekannt ist, wo diese grafische Ausgabe erzeugt werden soll.

Wurde im Rechenverfahren eine andere Form der grafischen Ausgabe generiert, lässt sie sich dennoch anzuzeigen. Hierzu gibt es die Variable: *\$name_bild*. Intern ist diese Variable, wenn ein Bild angezeigt werden soll, auf den Wert:

```
$namebild = $root_user_daten.$auth->auth["uid"]."/".$method."_".$random_name.".jpg";
```

gesetzt. Für einen speziellen Fall könnte der Wert wie folgt aussehen:

```
$namebild = 'd:/eping/daten/adminuser/holtrop_100000000.jpg';
```

Setzt man diese Variable selbst, kann somit jedes andere Bild angezeigt werden. (Anmerkung: das Bild muss nicht als JPG vorliegen, sondern kann jedes Format besitzen, das Browser anzeigen können)

6.2.3.5 Die Schnittstelle zum externen Rechenverfahren

Wie schon in der Einleitung dieses Kapitel beschreiben, wird intern das Rechenverfahren (sofern nicht anders vereinbart) immer automatisch angesprochen und ausgeführt. Dies ist nur dann möglich, wenn das Rechenverfahren bestimmte Voraussetzungen erfüllt:

- Die Kommunikation mit der Außenwelt über Eingabe- und Ausgabedatei(en), welche dem Programm in einer bestimmten Reihenfolge über die Kommandozeile übergeben werden. Konkret sieht der Aufruf wie folgt aus:
`<name_der_exe> <input_datei> <output_datei> <output_bild> <output_log>`
- dabei sind die letzten beiden Parameter optional

Nun kann nicht jedes Rechenverfahren genau dies gewährleisten, bzw. wäre der Aufwand zu groß. Für diesen Fall kann der Aufruf des Rechenverfahrens auch direkt erfolgen. Hierzu wird intern getestet, ob die Funktion:

```
function programm_ausfuehren($dateiexe,$dateixml,$dateixml_out,$dateips,$dateioutput_log)
{...}
```

vereinbart wurde. Ist dies der Fall, wird diese anstelle des automatischen Aufrufes des Rechenverfahrens angesprochen. Danach muss die komplette Funktionalität übernommen werden. In der Regel wird dies auf ein eigenständiges Ansteuern des Rechenverfahrens hinauslaufen. Ein Beispiel:

```
function programm_ausfuehren($dateiexe,$dateixml,$dateixml_out,$dateips,$dateioutput_log)
{
    global $my_exec;
    global $root_exe,$methode_name;
    $dateixml=str_replace('/', '\\',$dateixml);
    $dateixml_out=str_replace('/', '\\',$dateixml_out);
    $dateioutput_log=str_replace('/', '\\',$dateioutput_log);
    $ret=$my_exec->exec($dateiexe."                remote                ".$dateixml."                ".$dateixml_out."
".$dateioutput_log,$output,$result);
    $error=check_return_code($ret,$dateioutput_log);
```

```
if (!$error)
{
  global $bild_x,$bild_y,$bild_breite,$bild_hoehe;
  $bild_x=120;
  $bild_y=120;
  $bild_breite=605;
  $bild_hoehe=830;
  $post_prozessor=$root_exe.$methode_name."/show_abl.exe";
  $my_exec->exec($post_prozessor." ".$dateixml_out." ".$dateips,$output,$result);
}
return $error;
}
```

Zur Erläuterung:

- Zunächst werden die vier Standardparameter an die Funktion übergeben.
- Danach werden drei globale Variablen sichtbar gemacht. Im einzelnen sind dies:
 - `$my_exec` – diese Klasse übernimmt das Ansteuern von externen Programmen über die Kommandozeile
 - `$root_exe` – in dieser Variable ist der Pfad '`<...>/eping/exe`' gespeichert
 - `$methode_name` – in dieser Variable ist der Name des Rechenverfahrens gespeichert (zBsp: 'holtrop')
- In den nächsten drei Arbeitsschritten wird der Pfad der Eingabe- und der Ausgabedatei(en) angepasst, da das Rechenverfahren nur den Windows und nicht den Linux Standard versteht. Hierbei wird jedes Vorkommen von '/' durch '\\' ersetzt. (Anmerkung es sind zwei \ notwendig, da der erste \ nur anzeigt, dass nun ein „Sonderzeichen“ folgt.)
- Nun erfolgt der eigentliche Programmaufruf. Gut zu sehen ist, dass ein zusätzliches 'remote' in der Kommandozeile notwendig ist.
- Nach dem Aufruf des Rechenverfahrens, wird der Rückgabecode durch den Funktionsaufruf `check_return_code` überprüft, welcher ggf. eine Fehlermeldung ausgibt (diese Funktion ist spezifisch für das Beispielprogramm und muss für jedes Rechenverfahren erstellt werden)
- Ist kein Fehler aufgetreten wird nun durch den Aufruf eines Postprozessors ein PostScript-Bild erzeugt. Dabei wird für das spätere Umwandeln dieses Bildes nach JPG noch die X,Y Koordinaten, sowie die Bildhöhe und Bildbreite gesetzt.

6.2.4 Einbinden eines externen Programms (Non-XML)

Hier ist es notwendig, die XML-Daten vor dem Senden zum Programm in das eigene Format zu transformieren und das Ergebnis des Rechenverfahrens in eine resultierende XML-Datei umzuwandeln. Hierzu wird bei der Abarbeitung getestet, ob die Funktion:

```
function programm_ausfuehren($dateiexe,$dateixml,$dateixml_out,$dateips,$dateioutput_log)
{..}
```

existiert (siehe Kapitel 6.2.3.5). Durch das Erstellen einer solcher Funktion und dem eigenständigen Ansprechen des Rechenverfahrens kann auch eine Non-XML-Struktur verarbeitet werden. Hierzu muss zunächst die Standard-XML-Eingabe in das entsprechende Format umgewandelt werden und die Non-XML-Ausgabe im Anschluss in die entsprechenden XML-Ausgangsdaten überführt werden.

6.2.5 Erweiterung für das nicht interaktive Ansprechen von Rechenverfahren

Unter Umständen ist es notwendig, dass andere Programme Remote auf die im System integrierten Rechenverfahren zugreifen wollen. Um dieses zu gewährleisten, muss es eine Schnittstelle geben, welche die Eingabedaten für ein Rechenverfahren annimmt, kontrolliert, an das Rechenverfahren weiter leitet und die Ergebnisse zurückliefert, ohne das zwischendurch interaktiv eingegriffen werden muss.

Ein solches Skript ist im System integriert. Zurzeit wurde diese Erweiterung für ein Programm (Holtrop) bereitgestellt. Zu finden ist die Vorgehensweise in dem Skript:

```
.../www/individual/mtg/contact.php
```

Hier muss das entsprechende Rechenverfahren eingebunden werden. Außerdem übernimmt das Skript die Kontrollfunktion für die erlaubten Zugriffe. Sollen bestimmte Rechenverfahren nur einzelnen Usern erlaubt werden, müsste das Skript dementsprechend angepasst werden. Näheres finde Sie hierzu im Kapitel 4.2.

Achtung:

Intern wird beim Ausführen des Skriptes getestet, ob der User das Recht besitzt, eine solche nicht interaktive Anfrage zu stellen. Hierzu muss dem entsprechenden User in der Datenbank 'user_typ' =1 gesetzt werden.

6.2.6 Einbinden von Rechenverfahren in den individuellen Bereich

Um ein Skript im individuellen Bereich einzubinden, muss es zunächst wie für den registrierten Bereich aufgearbeitet werden. Doch anstatt die Datei <name>.php im Verzeichnis .../www/registered/<name> zu erzeugen, wird diese Datei außerhalb der Web-Freigabe erzeugt und direkt über das Mitarbeiterinterface in das System hinein geladen. (Registered Area → Mitarbeiter → Rechenverfahren für Kunden). Hierzu muss nur die öffnende und schliessende PHP-Tags entfernt werden (also ohne <?php und ?>!).

Beispiel:

```
page_open(array("sess" => "Example_Session", "auth" => "Example_Auth", "perm" => "Example_Perm"));

include("holtrop/holtrop_main.php");
include("holtrop/holtrop_error_table.phtml");

$sis_bild=1;
set_variablen("holtrop", "english", "upload", $sis_bild);
adjust_variable_if_load_old("holtrop", $sis_bild);

// HAUPTTEIL
programm_kopf();
programm_handle_upload($language, $userfile, $uploaded_file, $command);
programm_aufruf();
programm_enable_download($language, $dateibild, $dateips, $dateixml, $dateixml_out);
programm_fuss();

page_close();
```


6.2.7 Änderungen für das Einbinden von Rechenverfahren in den öffentlichen Bereich

Im öffentlichen Bereich fehlen nur die das PHP-Skript umschließende Funktionsaufrufe von `page_open(...)` und `page_close()`. Bis auf diese Tatsache erfolgt das Einbinden analog zum Einbinden eines Rechenverfahrens im registrierten Bereich.

6.2.8 Anpassungen bei Rechenverfahren mit sehr langer Ausführungszeit

Sollte ein Rechenverfahren eingebunden werden, welches eine sehr lange Ausführungszeit benötigt, um die Ergebnisse bereit zu stellen, müssen einige Anpassungen am System vorgenommen werden. Das wird immer dann notwendig, wenn bei einer Anfrage - bis zur Lieferung des Ergebnisses - ein Browser-Time-Out möglich ist.

Als Beispiel dient das Einbinden des Verfahrens Kelvin. Es benötigt bei entsprechender detaillierter Berechnung mehr als eine Stunde. So lange kann der Browser nicht auf das Ergebnis warten, weshalb dem Anwender statt dem Ergebnis ein Fortschrittbalken angezeigt wird. Diese HTML-Seite lädt sich automatisch in bestimmten Intervallen neu. (Hierzu muss der Client JavaScript aktiviert haben) Ist die Berechnung abgeschlossen wird dann das Ergebnis angezeigt. Um nicht die ganze Zeit den Browser offen zu lassen, gibt es bei diesem Rechenverfahren zusätzlich eine Verknüpfung, welche zur letzten (ggf. noch aktiven) Berechnung verweist. So kann jederzeit der Fortschritt bzw. das Ergebnis eingesehen werden. Die Fortschrittanzeige ist nur dann möglich, wenn das Rechenverfahren dafür eine Schnittstelle bereitstellt, z.B. durch eine oder mehrere Ausgabedatei(en).

Zusätzlich zu diesen Anpassungen muss gleichzeitig sichergestellt werden, dass jeder Anwender nur eine dieser Berechnungen gleichzeitig ausführt. Es sollte sichergestellt werden, dass nur eine Berechnung zu einem Zeitpunkt laufen kann. Dies ist aus zwei Gründen notwendig. Zum einen, damit Ergebnisse nicht vermischt werden (beim gleichen Anwender) und zum anderen um das System nicht zusätzlich zu belasten. Sollte eine Rechnung überdurchschnittlich lange benötigen, ist es recht wahrscheinlich, dass das System ausgelastet ist.

7 Individueller Bereich

Für die User können hier weitere Inhalte und Rechenverfahren bereitgestellt werden. Diese Inhalte können über eine automatisch generierte Seite im individuellen Bereich, oder über eine eigens dafür entworfenen 'Startseite' genutzt werden. Das Einstellen von diesen Objekten (HTML-Seiten, Bilder, Skripte ...) kann im Mitarbeiterbereich erledigt werden. Hierzu gibt es dort zwei Menüunterpunkte, welche ausgesucht werden können. In dem einem Bereich können Rechenverfahren eingebunden werden (vergleiche Kapitel 6.2.6) und im anderen Bereich jede Art von anderen Objekten. Intern wird zwischen folgenden Objektarten unterschieden:

- HTML (werden bei der Auswahl direkt angezeigt),
- Bilder (werden bei der Auswahl direkt angezeigt),
- PHP-Skripte (werden bei der Auswahl ausgeführt),
- Rechenverfahren (werden bei der Auswahl ausgeführt) und
- sonstige Objekte (werden bei der Auswahl zum Download angeboten).

Hierbei können natürlich die HTML-Seiten jedes der anderen Objekte einbinden bzw. dorthin verlinken.

individual / inhalte /				
		Name/Beschreibung	Rechte / Startseite	Datum
<input type="checkbox"/>		Kunde XYZ		
<input type="checkbox"/>		Mitarbeiter		
<input type="checkbox"/>		Programme		
<input type="checkbox"/>		Startseite Startseite.html	alle Mitarbeiter+ Rechte ändern/einsehen Startseite von Wuttke Startseite ändern	14:16 18 Nov 2003
<input type="checkbox"/>		test.php test.php	alle Mitarbeiter+ Rechte ändern/einsehen keine Startseite! Startseite ändern	12:20 05 Jan 2004
<input type="checkbox"/>		TestStartseite TestStartseite.html	alle Mitarbeiter+ Rechte ändern/einsehen keine Startseite! Startseite ändern	12:49 01 Jul 2004

Inhalt einstellen	Verzeichnis anlegen	Objekt(e) löschen
-------------------	---------------------	-------------------

Abbildung 5: Einstellen von Inhalten in den individuellen Bereich

7.1 Einstellen von Inhalten

Wie im vorherigen Kapitel geschrieben, erreicht man diesen Bereich über den Menüunterpunkt 'Inhalte für Kunden' im Bereich 'Mitarbeiter'. Im Bereich werden drei Optionen angeboten:

- Verzeichnisse anlegen

Durch das Anlegen von Verzeichnissen ist es möglich, eine Struktur zu erzeugen, so dass zum Beispiel alle Inhalte für einen Kunden in einem Verzeichnis hinterlegt werden können.

- Inhalte einstellen

Hier können die Inhalte eingestellt werden. Hierzu wählt man die entsprechende Datei auf der lokalen Festplatte aus, ordnet sie der korrekten Kategorie zu und gibt ggf. eine Beschreibung für diesen Inhalt an. Diese Beschreibung taucht bei der automatischen Generierung der Startseite als Bezeichner für die Verlinkung auf. Ist keine Beschreibung angegeben, wird automatisch der Dateiname übernommen.

- Inhalte löschen

Hier können veraltete Inhalte gelöscht werden.

- Zugriffsrechte anpassen

Hier kann festgelegt werden, welcher Kunde welche Dokumente einsehen kann.

- Startseite festlegen

Durch diese Option kann der entsprechende Inhalt einem Kunden als Startseite im individuellen Bereich zugeordnet werden.

Das Einstellen von Rechenverfahren entspricht dem Einstellen von Inhalten.

7.2 Auf die Inhalte zugreifen

Der Zugriff auf die eingestellten Inhalte erfolgt über `http://<rechnername>/individual/inhalte.phtml/` gefolgt von der Verzeichnisstruktur und dem Namen der Originaldatei. Der entsprechende Link findet auch in der Tabelle unter 'Namen/Beschreibung' Verwendung.

8 Sicherheit

8.1.1 Schutz der Daten

Es muss gewährleistet werden, dass auf dem Rechner kein Unbefugter Zugriff nehmen kann. Dabei ist insbesondere der Remote-Zugriff so zu beschränken, dass außer auf die ePING Browserseite kein weiterer Zugriff möglich ist. Hierzu sollte der Server in einem sicheren Netzwerk eingebettet sein. Zusätzlich sollte die Datenbank nur lokale Zugriffe erlauben, um die Sicherheit weiter zu erhöhen.

Der zweite sicherheitsrelevante Aspekt „Zugriff nur auf erlaubte Bereiche von ePING, sowie nur Einsicht in die eigenen Daten“ erfüllt ePING selbst. Um dies zu erreichen, kann jede einzelne PHP-Seite von ePING einzeln geschützt werden. Verwendung finden die freie Erweiterung von PHP um die PHP-LIB.

8.1.2 Schutz bei der Datenübermittlung

Um einen Schutz bei der Datenübermittlung zu gewährleisten, muss der entsprechende Apache-Server mit SSL-Unterstützung eingerichtet sein. Danach kann ePING im SSL-Modus betrieben werden. Zum Einschalten oder Ausschalten der SSL-Unterstützung muss der entsprechende Eintrag in der `defines.php` angepasst werden.