FormFactory

Template based web forms

Wolfgang Friebel

Motivation

- CGI scripts are often used to collect information and send it by email or store the data in a database
- Frequently written by inexperienced users or copied from Internet archives
- Usually not thoroughly checked for security holes
- Scripts tend to remain unmaintained, bugs stay in the code as usage frequency is usually low
- Need for well maintained code
- Need to help inexperienced CGI writers

A real world example

- At DESY a malfunctioning web form was reported by users in July this year
- One of the bugs found by K. Woller and corrected
- An analysis has shown that this code was derived from “Matts Script Archive” and used in many places at DESY
- Essential parts of the code are basically unchanged since 1997 when the script was downloaded initially
- The script has still the bug and the site claims “Downloaded over 2 million times since 1997” ...

A buggy part of the script

```
read(STDIN,$input,$ENV{'CONTENT_LENGTH'});

# convert + into space, %xx into ASCII char
$input=&trans($input);

(@pairs) = split(/\&/, $input);
foreach $p (@pairs) {
    ($name,$value) = split(/\=/,$p);
    $name{$name}=$value;
}
```

A buggy part of the script

```
read(STDIN,$input,$ENV{'CONTENT_LENGTH'});

# convert + into space, %xx into ASCII char
$input=&trans($input);

(@pairs) = split(/\&/, $input);
foreach $p (@pairs) {
    ($name,$value) = split(/\=/,$p ,2);
    $name{$name}=$value;
}
```

A buggy part of the script

```
read(STDIN,$input,$ENV{'CONTENT_LENGTH'});

# convert + into space, %xx into ASCII char
$input=&trans($input);      #convert %26 into &

(@pairs) = split(/\&/, $input); # wrong split
foreach $p (@pairs) {
    ($name,$value) = split(/\=/,$p ,2);
    $name{$name}=$value;
}
```

Conclusion

- Writing or adapting CGI scripts is simple
- Finding bugs in CGI scripts is much harder
  - do expect and handle arbitrary input
  - be prepared for failures related to network, ...
- Writing **secure** CGI scripts is very hard as well
  - unintended interactions with the OS by blindly passing input to the shell
  - checking for all possible side effects
- Do not try reinventing the wheel, reuse software

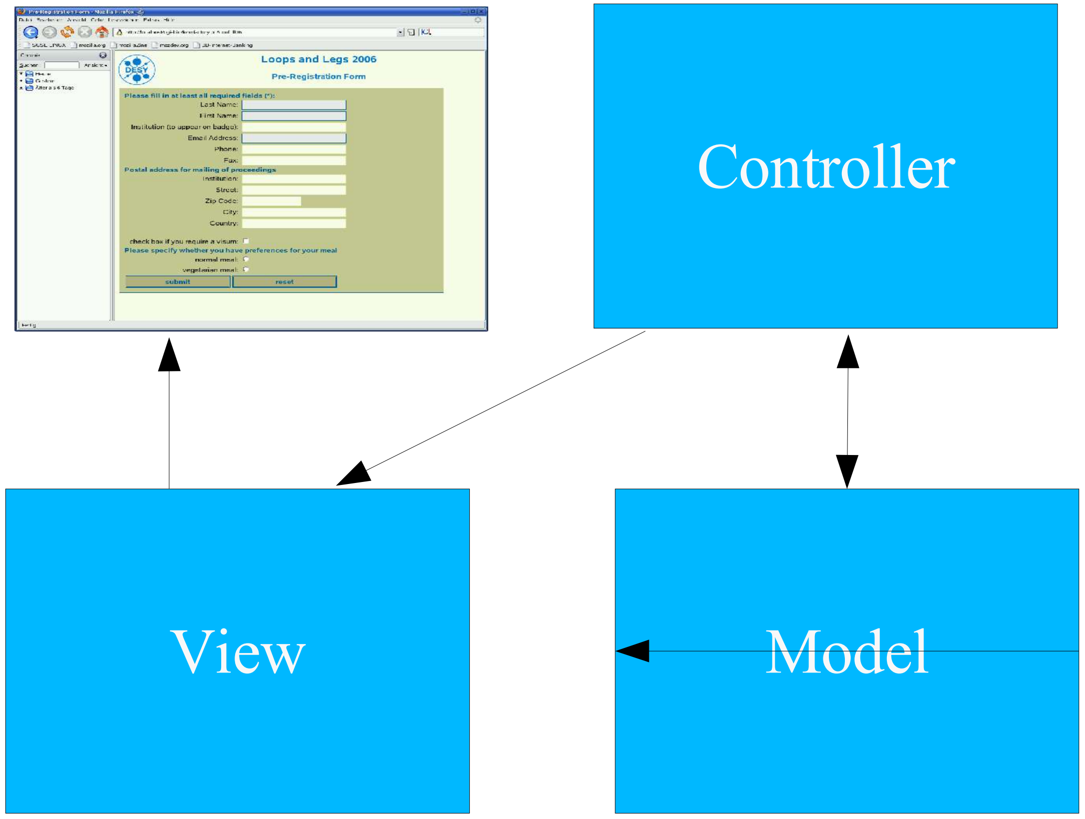
Available software

- Commercial web editors
  - can create web forms
  - but little support for postprocessing input
  - resulting script is monolithic, may rely on proprietary libraries, databases etc.
- Searching the Internet for solutions
  - overwhelming number of hits
  - many very simple scripts and commercial offers
  - no really good tool found
- Perl as a toolbox with thousands of modules
  - **CGI.pm** is de facto standard

Design principles

- Separation of program logic, data and layout as much as possible
- known as MVC (model, view, controller) design
- Use as much existing code as possible
- keep solution simple and modular
- Simple forms should not require code to be written
- Allow for complicated tasks

The Model View Controller Architecture



Loops and Legs 2006

Pre-Registration Form

Please fill in at least all required fields (*):

Last Name:
First Name:
Institution (to appear on badge):
Email Address:
Phone:
Fax:

Postal address for mailing of proceedings

Institution:
Street:
Zip Code:
City:
Country:

check box if you require a visum:

Please specify whether you have preferences for your meal

normal meal:
vegetarian meal:

1. send request from browser
2. controller gets info from model
3.model is stored in database
4. controller invokes view
5. view renders next screen

Database

taken from book "agile web development with rails"

Selection of tools: Program flow

- **CGI** as the underlying base class
- **CGI::Application** to split the task (CGI processing states), many plugins available
  - **AnyTemplate** plugin to be able using templates
    - could also use a specific plugin for a given templating system
  - **ValidateRM** plugin to enable form validation
    - is calling **Data::FormValidator**

CGI.pm and its subclasses

- should be used in all perl based CGI scripts
- free you from parsing HTTP messages
- give easy access to **script parameters**:

```
use CGI qw/:standard/;
@names = param();
$email = param('email');
```

- help you in **debugging** your script

```
use CGI::Carp qw(fatalsToBrowser);
```

- assist you in generating HTML **tag pairs**

```
print h1('Chapter 1');
```

Selection of tools: Data model

- describe the data using attribute hashes (name, type, ...)

```
payment => { label => 'Payment method',
             fieldtype => 'select',
             value => [ 'cash', 'visa' ],
             labels => { cash => 'Cash',
                         visa => 'Visa Card'},
             dbtype => 'varchar (5)' }, ...
```

- **DBI** module for storing the data in a broad range of available databases (mysql, Oracle, SQLite, ...)
- **Ima::DBI** for lazy loading and SQL encapsulation
- **Net::SMTP** for data transport by email

Advantages of using Ima::DBI

- delays opening of DB connection until required
- Guarantees only one DB connection per DB
    - important for persistent applications
- Only one prepared handle per SQL statement
- Encourages use of bind parameters and columns
- Helps keeping SQL statements in a central place
- Can do extra (taint) checks for input data

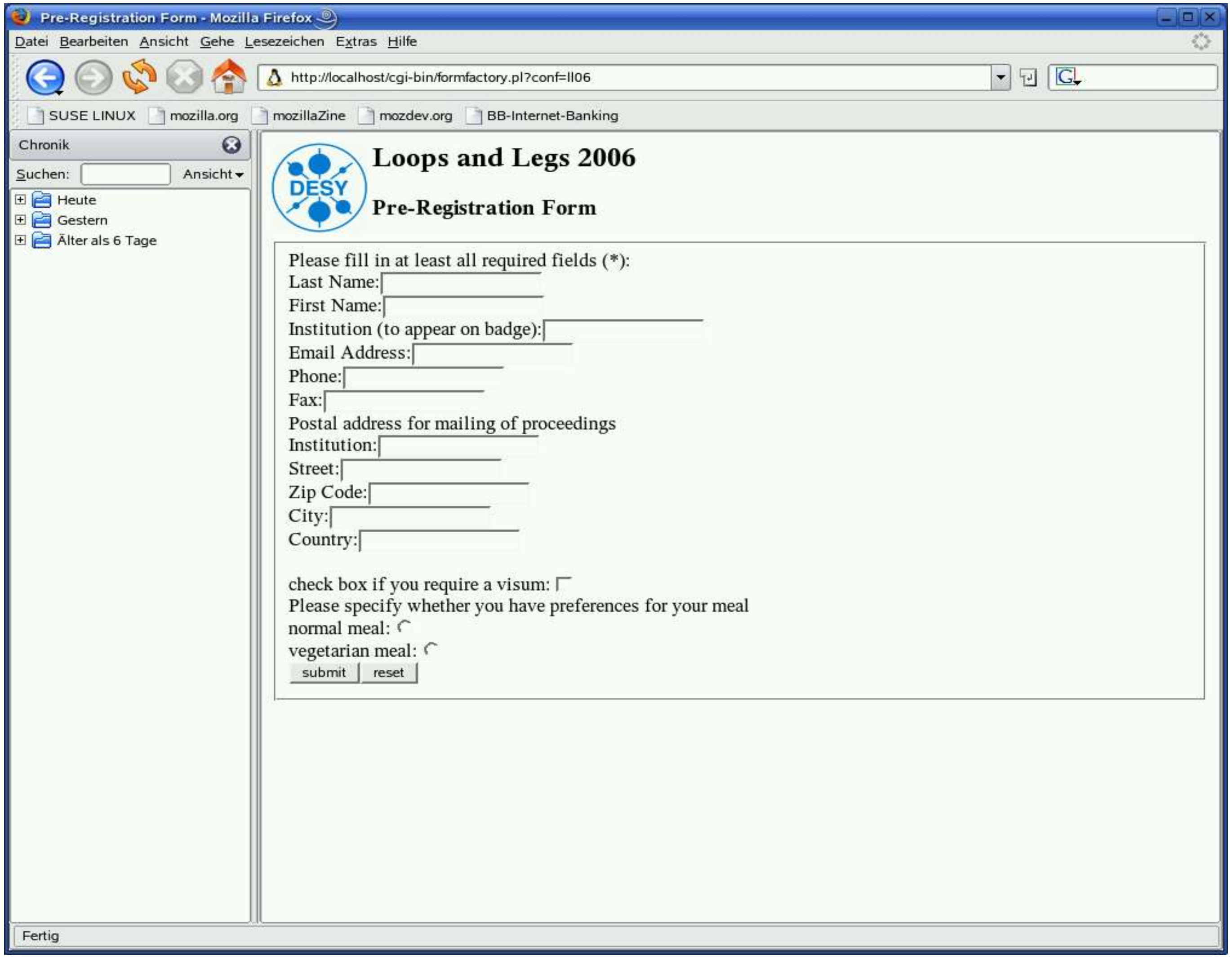
Sample Ima::DBI usage

```
package Foo;
      use base qw(Ima::DBI);
      # Set up database connections (but don't connect yet)
      Foo->set_db('Users', 'dbi:Oracle:Foo', 'admin', 'passwd');
      # Set up SQL statements to be used through out the program.
      Foo->set_sql('FindUser', <<"SQL", 'Users');
          SELECT  *
          FROM    Users
          WHERE   Name LIKE ?
      SQL

      package main;
          $obj = Foo->new;
          my $sth = $obj->sql_FindUser;   # Does connect & prepare
          $sth->execute('Fri%');          # bind_params & execute.
          @names = $sth->fetchall;
```

Selection of tools: Layout

- Use of cascading style sheets (**CSS**)
  - provide CSS sample files, as support for a wide range of browsers is tricky (Netscape4!)
  - style of page changes without touching its content
  - **Template::Toolkit** to write the HTML code
    - advantage: access to all data visible within CGI, even access to database
    - used in large projects (sympa mailing list manager)
    - other templating systems possible(**HTML::Template**, **Petal**, ...)

Sample page without CSS


Pre-Registration Form - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://localhost/cgi-bin/formfactory.pl?conf=ll06
SUSE LINUX
mozilla.org
mozillaZine
mozdev.org
BB-Internet-Banking
Chronik
Suchen:
Ansicht
Heute
Gestern
Älter als 6 Tage
DESY
Fertig

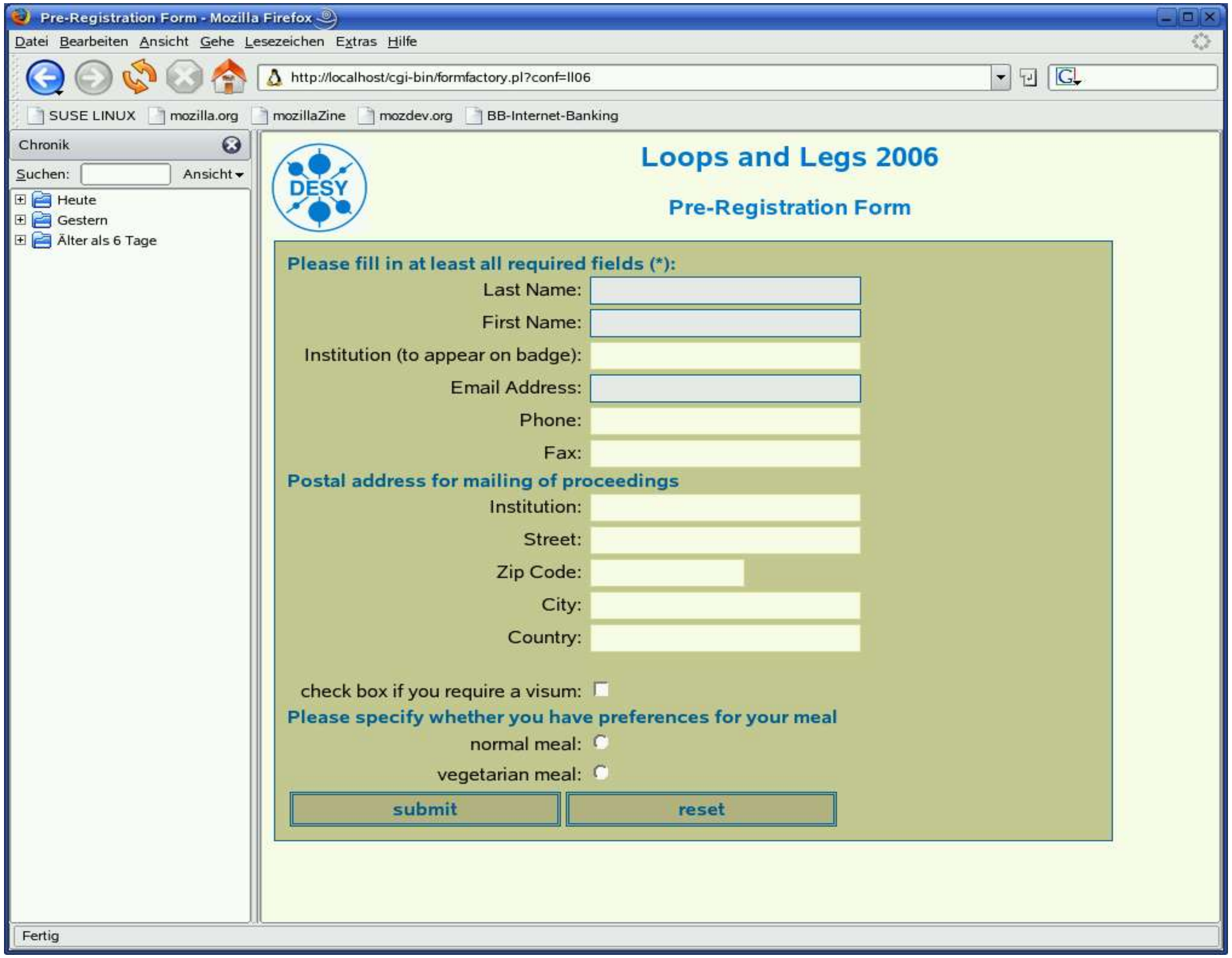
Loops and Legs 2006

Pre-Registration Form

Please fill in at least all required fields (*):
Last Name:
First Name:
Institution (to appear on badge):
Email Address:
Phone:
Fax:
Postal address for mailing of proceedings
Institution:
Street:
Zip Code:
City:
Country:

check box if you require a visum:
Please specify whether you have preferences for your meal
normal meal:
vegetarian meal:
submit reset

Sample page with CSS



Pre-Registration Form - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://localhost/cgi-bin/formfactory.pl?conf=ll06
SUSE LINUX
mozilla.org
mozillaZine
mozdev.org
BB-Internet-Banking
Chronik
Suchen:
Ansicht
Heute
Gestern
Älter als 6 Tage
DESY
Fertig

Loops and Legs 2006

Pre-Registration Form

Please fill in at least all required fields (*):

Last Name:	
First Name:	
Institution (to appear on badge):	
Email Address:	
Phone:	
Fax:	

Postal address for mailing of proceedings

Institution:	
Street:	
Zip Code:	
City:	
Country:	

check box if you require a visum: ☐

Please specify whether you have preferences for your meal

normal meal: ○

vegetarian meal: ○

submit | reset

The Template Toolkit (TT2)

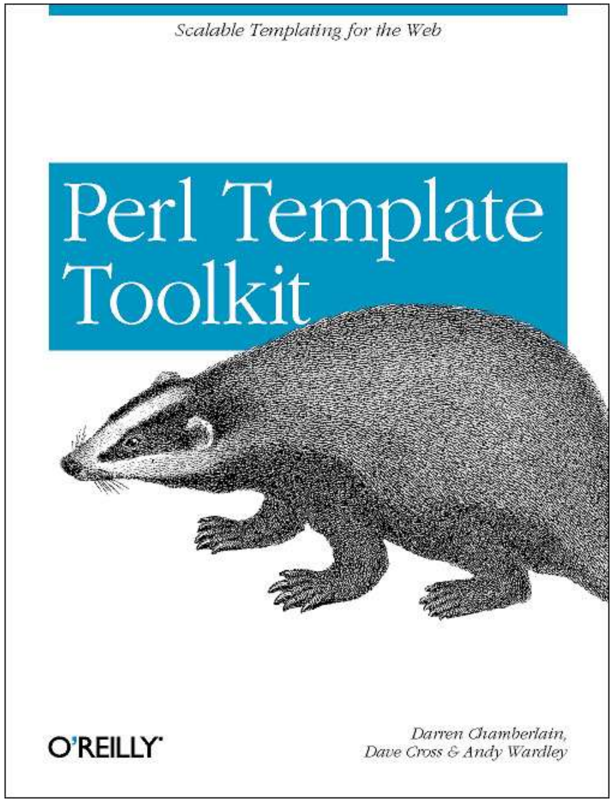
- very powerful and popular among the myriad of perl based templating systems (-> “badger book”)
- well suited to structure pages

  [% INCLUDE header %]

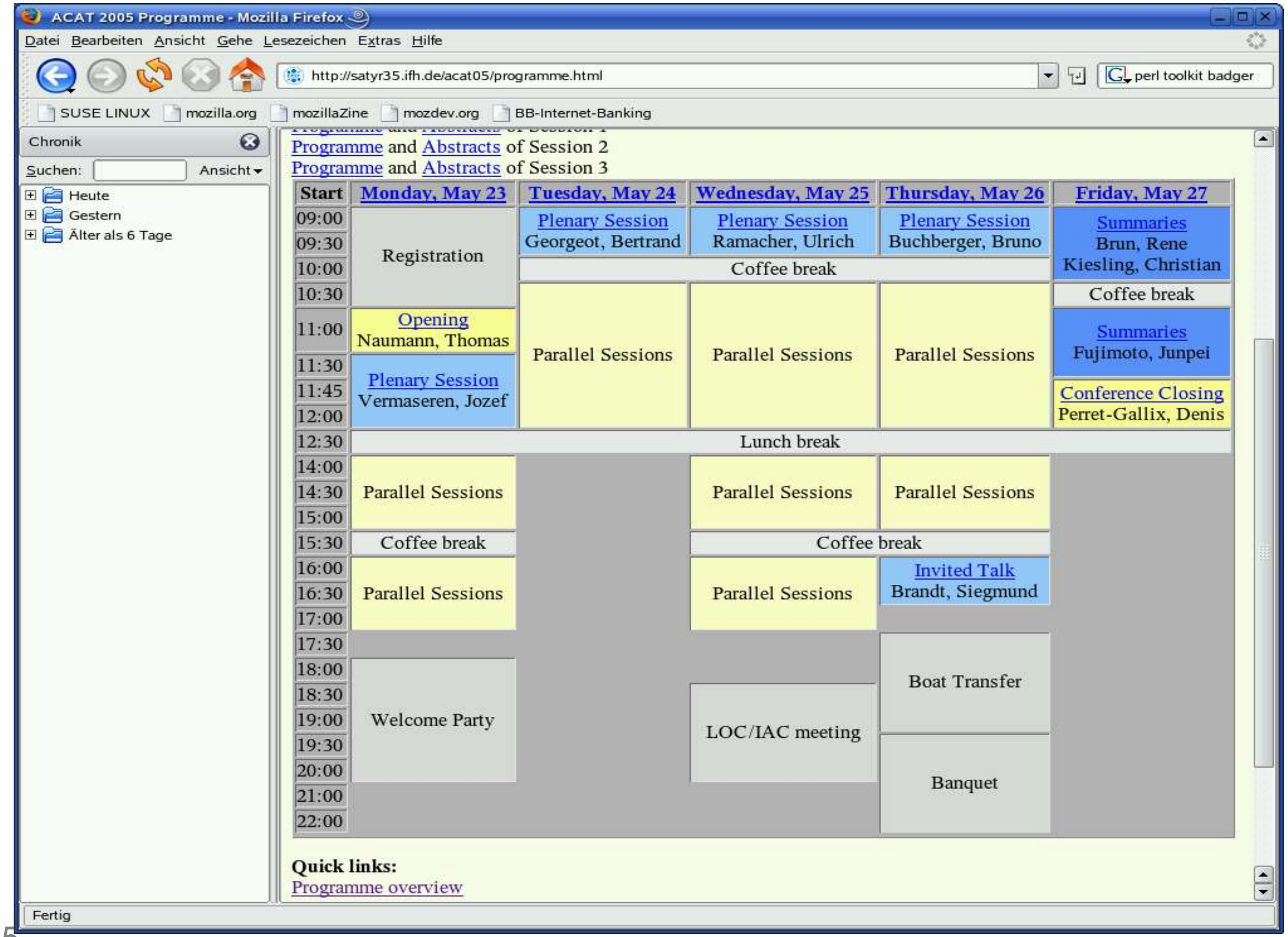
  [% INCLUDE menu %]

  [% INCLUDE maintext %]

  [% INCLUDE footer %]
- can pass variables to modify page contents
- can access DB's directly


Scalable Templating for the Web
Perl Template Toolkit
O'REILLY
Darren Chamberlain, Dave Cross & Andy Wardley

Sample page using Template Toolkit


ACAT 2005 Programme - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://satyr35.ifh.de/acat05/programme.html

perl toolkit badger

SUSE LINUX mozilla.org mozillaZine mozdev.org BB-Internet-Banking

Chronik

Suchen: Ansicht

- Heute
- Gestern
- Älter als 6 Tage

Programme and Abstracts of Session 2
Programme and Abstracts of Session 3

Start	Monday, May 23	Tuesday, May 24	Wednesday, May 25	Thursday, May 26	Friday, May 27
09:00	Registration	Plenary Session Georgeot, Bertrand	Plenary Session Ramacher, Ulrich	Plenary Session Buchberger, Bruno	Summaries Brun, Rene Kiesling, Christian
09:30					
10:00		Coffee break			
10:30		Parallel Sessions	Parallel Sessions	Parallel Sessions	Coffee break
11:00	Opening Naumann, Thomas				Summaries Fujimoto, Junpei
11:30	Plenary Session Vermaseren, Jozef				
11:45					Conference Closing Perret-Gallix, Denis
12:00					
12:30	Lunch break				
14:00	Parallel Sessions		Parallel Sessions	Parallel Sessions	
14:30					
15:00					
15:30	Coffee break		Coffee break		
16:00	Parallel Sessions		Parallel Sessions	Invited Talk Brandt, Siegmund	
16:30					
17:00					
17:30				Boat Transfer	
18:00	Welcome Party				
18:30			LOC/IAC meeting		
19:00					
19:30				Banquet	
20:00					
21:00					
22:00					

Quick links:
Programme overview

Fertig

Template toolkit sample code

```
The following data have been entered by you:
[%- USE CGI %]
[%- params = CGI.param() %]
[%- FOREACH par = params %]
  [%- NEXT IF par == 'submit' %]
  [%- NEXT IF NOT form.item.$par %]
  [%- NEXT IF form.item.$par.fieldtype == 'hidden' %]
  [%- IF form.item.$par.label %]
    [%- label = form.item.$par.label %]
  [%- ELSE %]
    [%- label = par %]
  [%- END %]
  [%- label %] = [% CGI.param(par) %]
[% END %]
```

Putting it all together: FormFactory

- work in progress (perl module)
- **FormFactory** integrates above mentioned pieces
- reads and parses a config file
- implements the workflow
  - form presentation
  - form validation
  - results presentation (in Web browser)
  - results postprocessing (store in DB, email)
- about 500 lines of code (without doc)
- directly calling perl modules containing about 30 000 LOC

The CGI script

- is to my knowledge bug free:

```
#!/usr/local/bin/perl
use FormFactory;
my $webapp = FormFactory->new();
$webapp->run();
```

- but could still be improved ;-)
  - use strict
  - use warnings
  - taint checks on

The config file

```
[general]
    title = Notebooks at DESY
    style = /computing/style2.css
    vardefs = notebook.def
[mail]
    mailto = wolfgang.friebel@desy.de, $email
    mailfrom = uco-zeuthen@desy.de
    mailsubject =  notebook mac address registration for $name
[db]
    dbname = mysql:test
    dbuser = wf
    dbpass = yyy
    dbtable = macs
[form]
...
```

Form definition

```
[form]
 next_runmode = process_template(results)
 title = Registration Form
 # mandantory fields do start with an asterix
 *name
 *firstname
 *email
 phone
 hostname
 # additional text in the form starts with a colon
 : Enter the MAC addreses in the form xx:xx:xx:xx:xx:xx for
one or two interfaces
 ethernet
 wlan
 # buttons are denoted by angle brackets
 <submit>
```

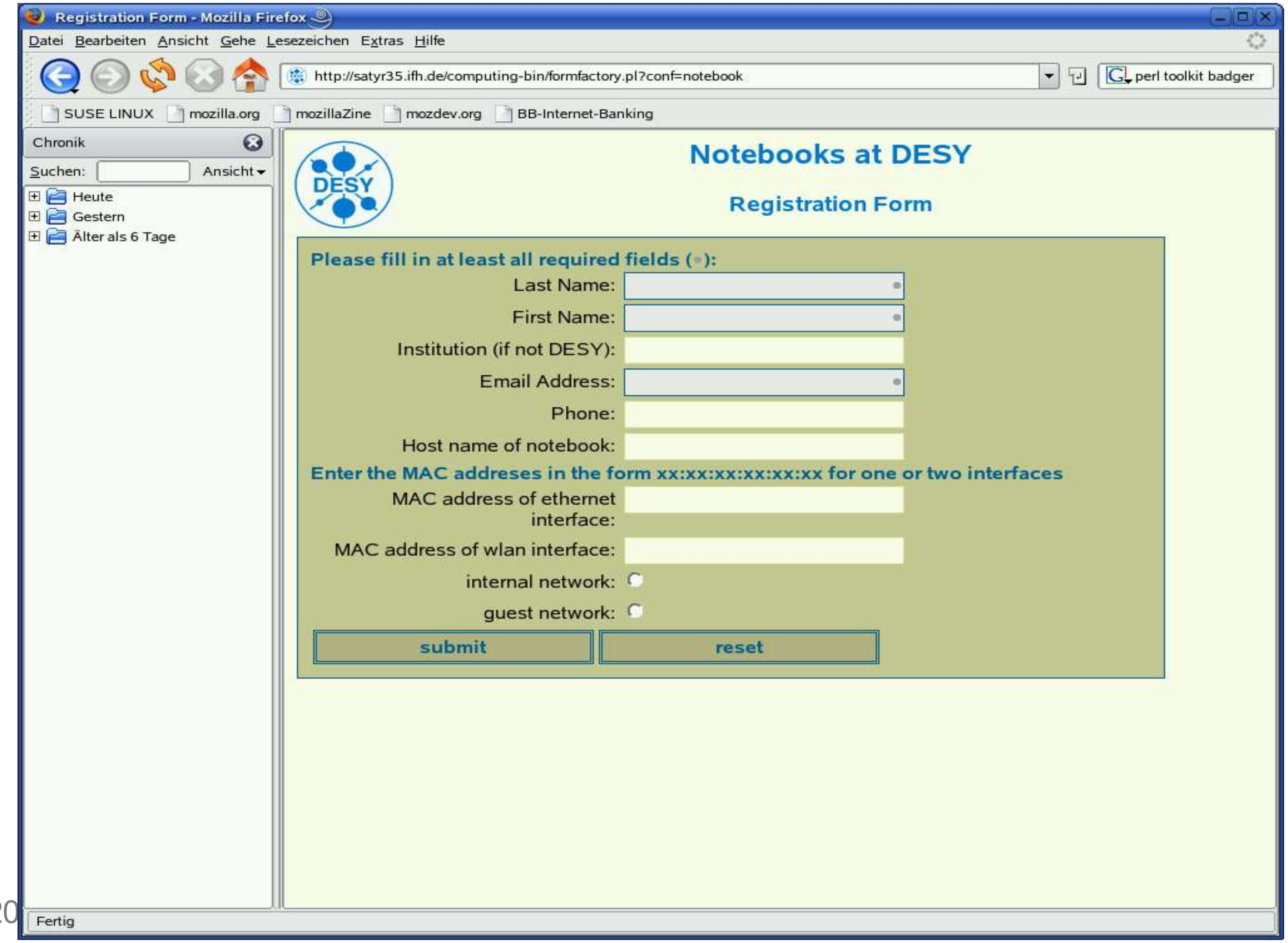
Workflow for Template processing

- not yet available (for the moment a fixed schema is used)

```
[results]
    next_runmode = process_template(mailresults)
    (results)
[mailresults]
    # the parameters from the [mail] section should be here
    next_runmode = update_db(storeresults)
    (letter)
[storeresults]
    # the parameters from the [db] section should be here
    dbtable = macs
```

- process_template and update_db are procedures called in FormFactory using CGI::Application

A sample form



Registration Form - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://satyr35.ifh.de/computing-bin/formfactory.pl?conf=notebook

Notebooks at DESY

Registration Form

Please fill in at least all required fields (•):

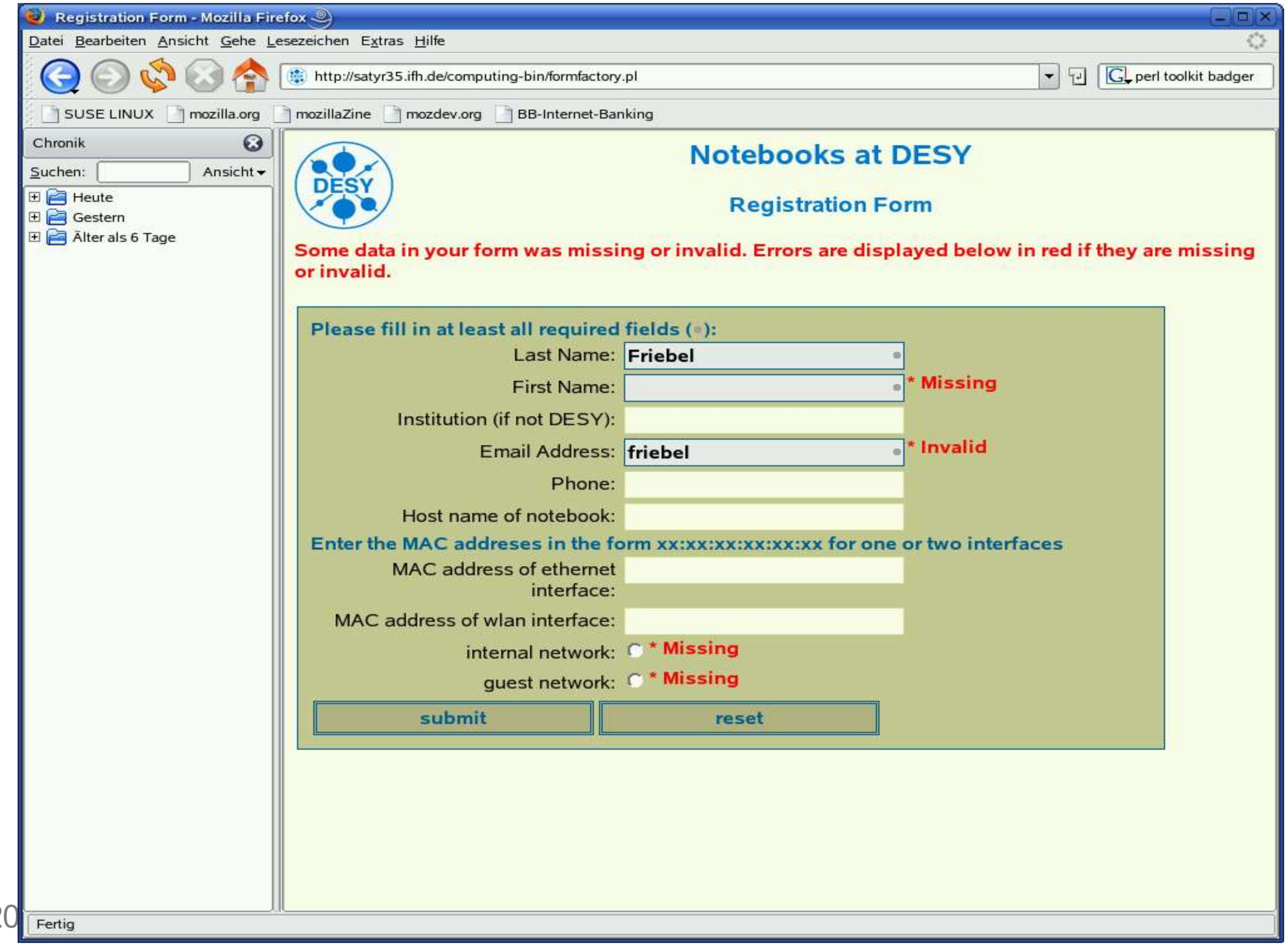
Last Name:	•
First Name:	•
Institution (if not DESY):	
Email Address:	•
Phone:	
Host name of notebook:	

Enter the MAC addreses in the form xx:xx:xx:xx:xx:xx for one or two interfaces

MAC address of ethernet interface:	
MAC address of wlan interface:	
internal network:	○
guest network:	○

submit reset

Entering incomplete and invalid data


Registration Form - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://satyr35.ifh.de/computing-bin/formfactory.pl
perl toolkit badger
SUSE LINUX mozilla.org mozillaZine mozdev.org BB-Internet-Banking
Chronik
Suchen: Ansicht
Heute
Gestern
Älter als 6 Tage
DESY
Fertig

Notebooks at DESY

Registration Form

Some data in your form was missing or invalid. Errors are displayed below in red if they are missing or invalid.

Please fill in at least all required fields (●):

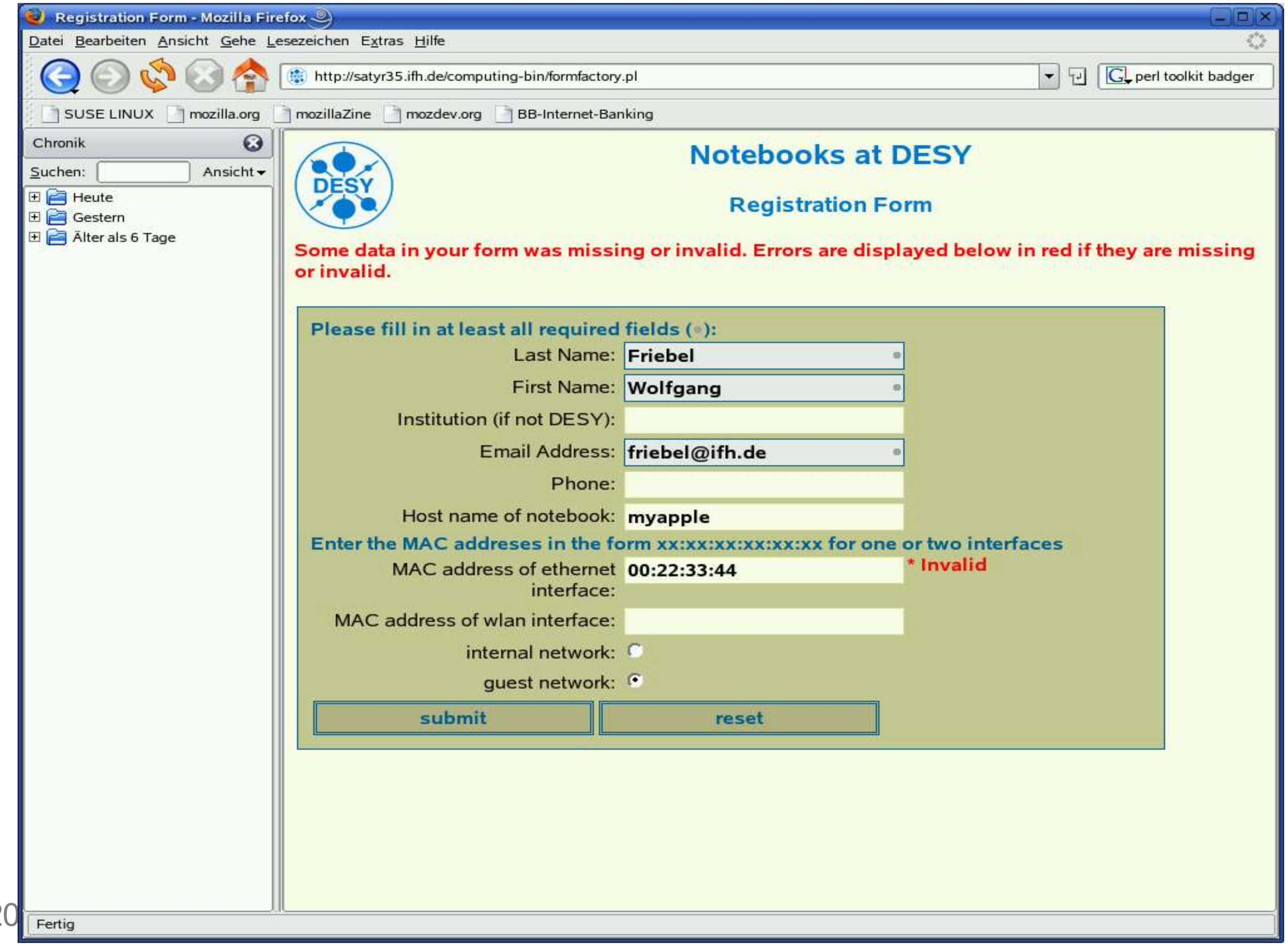
Last Name:	Friebel
First Name:	* Missing
Institution (if not DESY):	
Email Address:	friebel * Invalid
Phone:	
Host name of notebook:	

Enter the MAC addreses in the form xx:xx:xx:xx:xx:xx for one or two interfaces

MAC address of ethernet interface:	
MAC address of wlan interface:	
internal network:	○ * Missing
guest network:	○ * Missing

submit reset

Entering an invalid MAC address


Registration Form - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://satyr35.ifh.de/computing-bin/formfactory.pl
perl toolkit badger
SUSE LINUX mozilla.org mozillaZine mozdev.org BB-Internet-Banking
Chronik
Suchen: Ansicht
Heute
Gestern
Älter als 6 Tage
Fertig

Notebooks at DESY

Registration Form

Some data in your form was missing or invalid. Errors are displayed below in red if they are missing or invalid.

Please fill in at least all required fields (●):

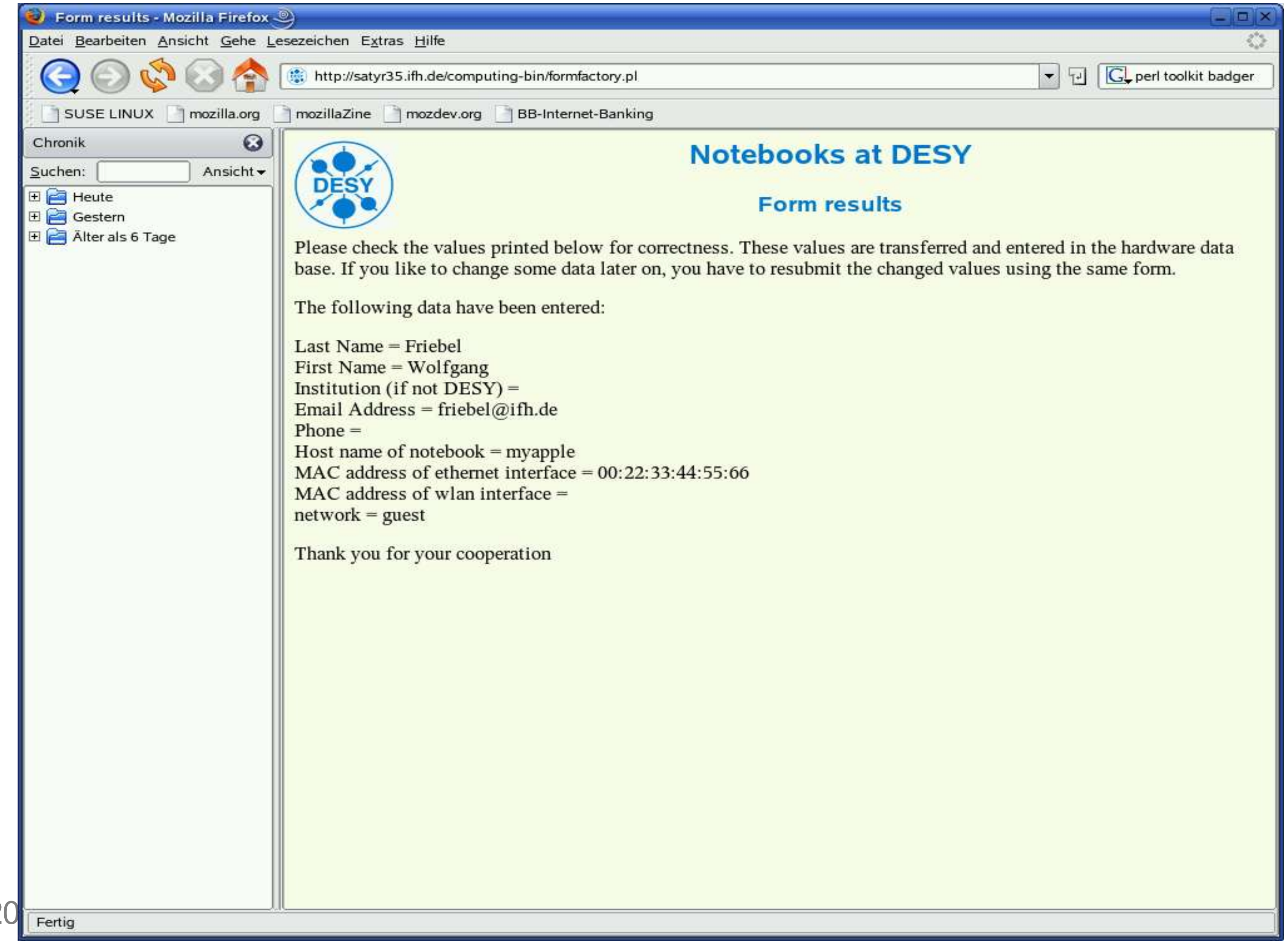
Last Name:	Friebel ●
First Name:	Wolfgang ●
Institution (if not DESY):	
Email Address:	friebel@ifh.de ●
Phone:	
Host name of notebook:	myapple

Enter the MAC addreses in the form xx:xx:xx:xx:xx:xx for one or two interfaces

MAC address of ethernet interface:	00:22:33:44 * Invalid
MAC address of wlan interface:	
internal network:	○
guest network:	●

submit reset

Results on screen



Notebooks at DESY

Form results

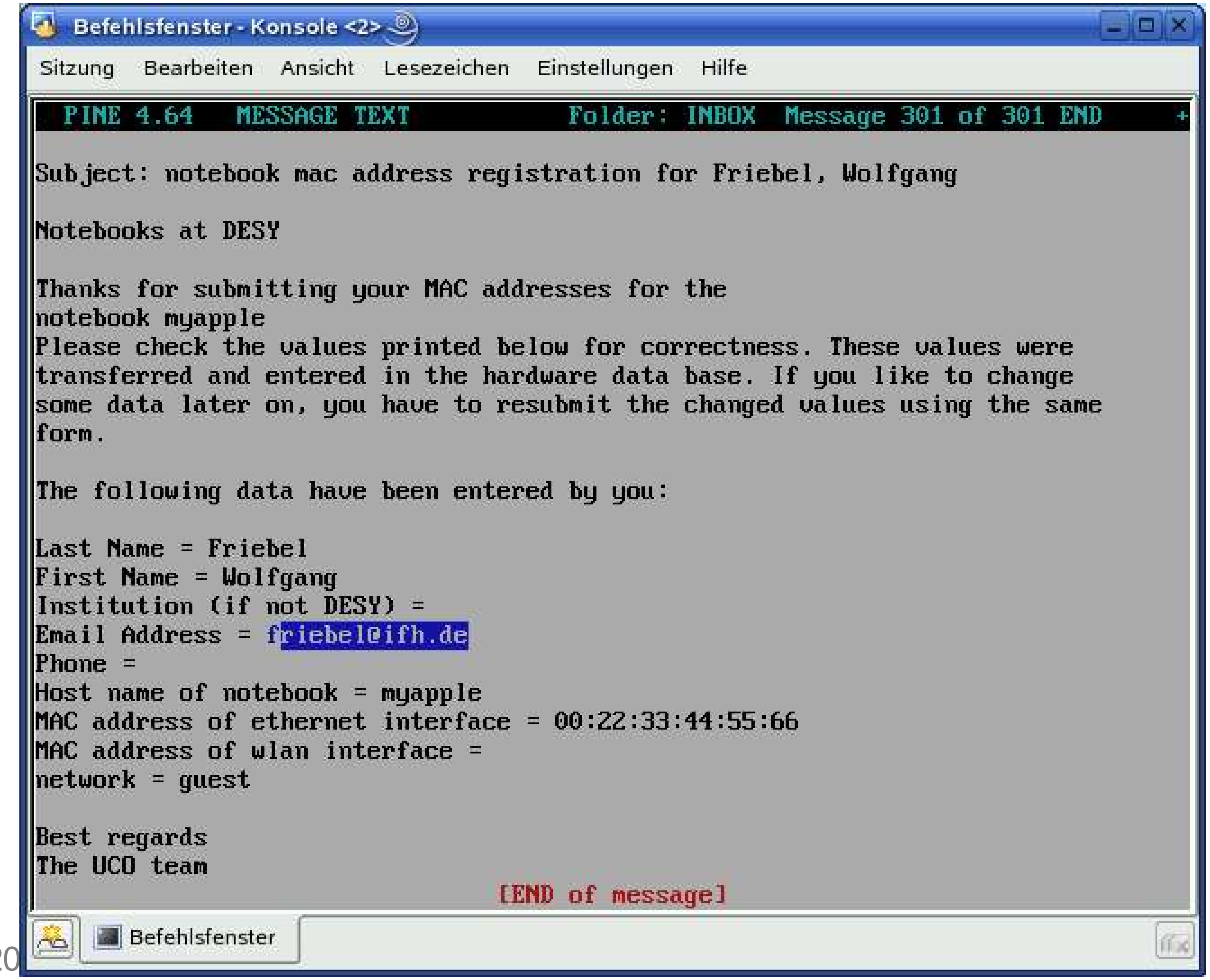
Please check the values printed below for correctness. These values are transferred and entered in the hardware data base. If you like to change some data later on, you have to resubmit the changed values using the same form.

The following data have been entered:

Last Name = Friebel
First Name = Wolfgang
Institution (if not DESY) =
Email Address = friebel@ifh.de
Phone =
Host name of notebook = myapple
MAC address of ethernet interface = 00:22:33:44:55:66
MAC address of wlan interface =
network = guest

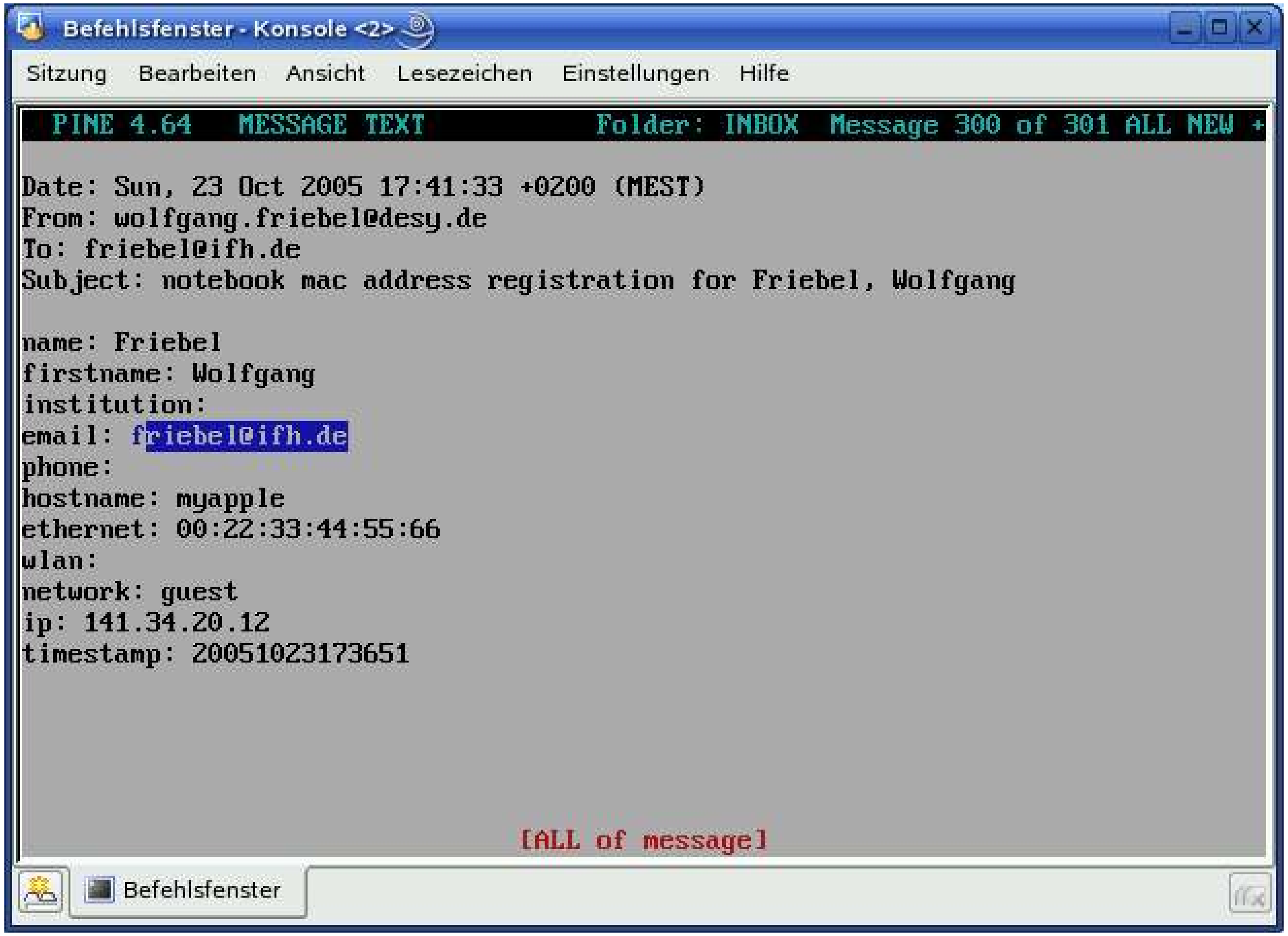
Thank you for your cooperation

Results as Email to User


Befehlsfenster - Konsole <2>
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
Befehlsfenster

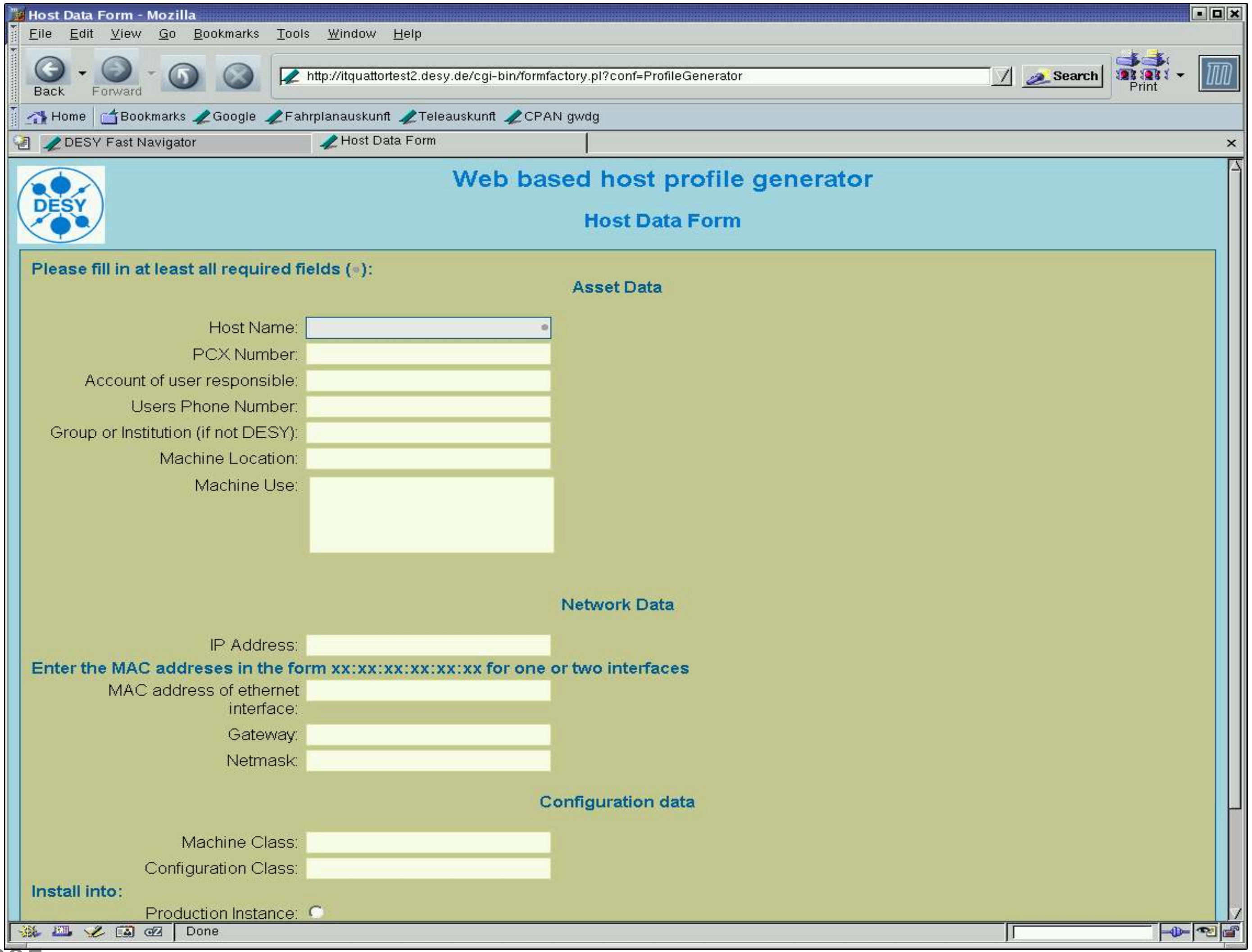
```
  PINE 4.64   MESSAGE TEXT          Folder: INBOX  Message 301 of 301 END

Subject: notebook mac address registration for Friebel, Wolfgang

Notebooks at DESY

Thanks for submitting your MAC addresses for the
notebook myapple
Please check the values printed below for correctness. These values were
transferred and entered in the hardware data base. If you like to change
some data later on, you have to resubmit the changed values using the same
form.

The following data have been entered by you:

Last Name = Friebel
First Name = Wolfgang
Institution (if not DESY) =
Email Address = friebel@ifh.de
Phone =
Host name of notebook = myapple
MAC address of ethernet interface = 00:22:33:44:55:66
MAC address of wlan interface =
network = guest

Best regards
The UCO team
                              [END of message]
```

Results to Maintainer


Befehlsfenster - Konsole <2>
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
Befehlsfenster

```
  PINE 4.64   MESSAGE TEXT          Folder: INBOX  Message 300 of 301 ALL NEW +

Date: Sun, 23 Oct 2005 17:41:33 +0200 (MEST)
From: wolfgang.friebel@desy.de
To: friebel@ifh.de
Subject: notebook mac address registration for Friebel, Wolfgang

name: Friebel
firstname: Wolfgang
institution:
email: friebel@ifh.de
phone:
hostname: myapple
ethernet: 00:22:33:44:55:66
wlan:
network: guest
ip: 141.34.20.12
timestamp: 20051023173651



                                [ALL of message]
```

Used already elsewhere


Host Data Form - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward
Search
Print
Home Bookmarks Google Fahrplanauskunft Teleauskunft CPAN gwdg
DESY Fast Navigator
Host Data Form
DESY
Done

http://itquattortest2.desy.de/cgi-bin/formfactory.pl?conf=ProfileGenerator

Web based host profile generator

Host Data Form

Please fill in at least all required fields (●):

Asset Data

Host Name: ●
PCX Number:
Account of user responsible:
Users Phone Number:
Group or Institution (if not DESY):
Machine Location:
Machine Use:

Network Data

IP Address:

Enter the MAC addresses in the form xx:xx:xx:xx:xx:xx for one or two interfaces

MAC address of ethernet interface:
Gateway:
Netmask:

Configuration data

Machine Class:
Configuration Class:

Install into:

Production Instance: ○

Summary

- perl offers a huge amount of high quality modules
  - designing powerful applications gets easier
  - can be difficult to find the right modules from the almost 9000 ones offered
  - some of the modules used here are fairly new or got popular only recently
- FormFactory used up to now for
  - workshop registration
  - MAC address registration of notebooks
- Different people responsible for parts of the task
  - DB, CSS, web content, processing of data

Outlook

- to do: better definition of workflow in the config file
- Ease access to database using **Class::DBI**
- Do support more form elements
- Provide more templates for common tasks
- Try to use **Catalyst**, a very poweful MVC framework in perl (catalyst.perl.org)
  - even less code to write
  - much more integrated and rapidly evolving
- More fine grained control by using **CGI::Ajax**
  - Asynchronously call Javascript using XML

What others do with the web

- Web 2.0
- At recent EuroOSCON many people reported to work on AJAX or to have it in production
- most popular combo: php+ajax
- popular MVC frameworks
  - in Java: Struts, Tapestry
  - in Ruby: Ruby on Rails