

# *PHP2 -- CoMa*

Im Rahmen des Fortgeschrittenenpraktikums  
programming-in-the-many

CAU Kiel  
Wintersemester 2004 / 2005

# *Anforderungen an CoMa*

- ◆ Co(nference )Ma(naging Tool)
- ◆ CoMa soll
  - ◆ den Vorablauf einer Konferenz vereinfachen
  - ◆ die Verwaltung der Anmeldungen übernehmen
  - ◆ die Verwaltung der Beiträge übernehmen
  - ◆ Teilnehmern notwendige Informationen bieten
  - ◆ die Beiträge sinnvoll an die Reviewer verteilen
  - ◆ eine Vorauswahl der paper vorschlagen

# *Was CoMa bietet*

- ◆ CoMa bietet eine dynamische Ansicht der Inhalte für beliebig viele Konferenzen
- ◆ Hierbei erkennt das System die Rolle des Teilnehmers in einer Konferenz und bietet ihm alle Aktionen, die er ausführen darf, konferenzübergreifend und geordnet nach Rollen
- ◆ Je nach Tätigkeit wechselt das System den Benutzer automatisch in die benötigte Rolle
- ◆ Aktionen, die der Benutzer nicht ausführen darf, werden ihm nicht angeboten

# *Ablauf einer Konferenz mit CoMa*

- ◆ Aufsetzen des Systems
- ◆ Ansetzen einer Konferenz
- ◆ Call-for-Papers
- ◆ Zusammenstellung eines Bewertungskomitees
- ◆ Anmeldung von Autoren
- ◆ Papereinsendungen
- ◆ Bewertungen der Beiträge
- ◆ Zusammenstellung eines Programms
- ◆ Benachrichtigung der Autoren
- ◆ Beginn der eigentlichen Konferenz

# *Moduleinteilung*

## ◆ Systemseite

- ◆ Installation
- ◆ Administration
- ◆ Funktionsverwaltung
- ◆ Forum

## ◆ Rollenseite

- ◆ Chair
- ◆ Reviewer
- ◆ Author

# *Installation*

- ◆ Voraussetzungen
  - ◆ „genügend“ Webspace
  - ◆ eine laufende PHP - Installation
  - ◆ MySQL – Datenbank
  - ◆ Zugang zu einem FTP – Server
  
- ◆ Ablauf
  - ◆ kopieren des CoMa-Pakets auf den Webspace
  - ◆ aufrufen der „index.php“
  - ◆ befolgen der Anweisungen
    - ◆ Eingabe der nötigen Zugangsdaten
    - ◆ Einrichtung des Administrators
  - ◆ löschen des Ordners „Install“

# *Funktionen des Administrators*

- ◆ Der Administrator hat folgende Aufgaben
  - ◆ Erstellen von Konferenzen
  - ◆ Einrichten des Chairaccount
  - ◆ Löschen der Konferenzen
  - ◆ Korrektur der Zugangsdaten
    - ◆ des FTP – Servers
    - ◆ der Datenbank
  - ◆ Ausserdem kann er einige Daten ändern
    - ◆ Name und Beschreibung der Konferenz
    - ◆ Webauftritt der Konferenz

# *Funktionen des Chairs*

## **Der Chair übernimmt die Leitung der Konferenz**

- ◆ Er kann sämtliche Daten der Konferenz ändern
- ◆ Er legt Themen und Ablauf der Konferenz fest
- ◆ Er kennt als einziger alle Teilnehmer und kann ihre Daten einsehen
- ◆ Er hat den vollständigen Überblick über alle paper
- ◆ Er verwaltet die paper
- ◆ Paperverwaltung
  - ◆ Der Chair kann einzelne paper einem bestimmten Reviewer vorlegen
  - ◆ Er muss ein paper letztendlich zur Konferenz zulassen oder es ablehnen
- ◆ Benutzerverwaltung
  - ◆ Der Chair lädt einen Benutzer für eine bestimmte Rolle ein
  - ◆ Ebenso kann er ihm die Rolle wieder entziehen

# *Funktionen des Authors*

- ◆ Durch das Anlegen einer Konferenz wird automatisch auf der Loginseite der Call-for-Papers gestartet
- ◆ Ein Interessierter kann sich daraufhin als Autor anmelden
- ◆ Er erhält dann automatisch das Recht, Beiträge einzureichen
- ◆ Ein Autor kann
  - ◆ zu jeder Konferenz, bei der er sich anmeldet beliebig viele paper einreichen
  - ◆ diese paper editieren bis die paper-submission-deadline abgelaufen ist
  - ◆ jederzeit sein paper löschen und es so von der Konferenz zurückziehen

# *Funktionen des Reviewers*

- ◆ Der Reviewer bewertet ihm zugeteilte paper und bestimmt damit nicht unwesentlich das Programm der Konferenz
- ◆ Im Vorfeld seiner Tätigkeit kann er angeben, mit welchen der Themen der Konferenz er sich bevorzugt beschäftigen möchte
- ◆ Ausserdem kann er einzelne paper zum bewerten anfordern
- ◆ Natürlich hat er die Möglichkeit, ihm zugeteilte paper abzulehnen
- ◆ Bis die Review-Deadline abgelaufen ist, kann eine Bewertung beliebig geändert werden
- ◆ Ebenso können abgelehnte paper doch noch bewertet werden

# *Paperverteilung*

- ◆ Der PTR-Algorithmus verteilt die paper zur Bewertung an die Reviewer, dabei
  - ◆ hält er nach Möglichkeit die vorgegebene Mindestanzahl von Reviewern ein
  - ◆ achtet er auf eine möglichst ausgewogene Verteilung
  - ◆ berücksichtigt er bevorzugte Themen der Reviewer
  - ◆ beachtet er reviewer, die bestimmte paper nicht bewerten dürfen
- ◆ Sollte ein Reviewer ein ihm zugeteiltes paper ablehnen, verteilt der PTR es an weitere Reviewer um die Mindestanzahl einzuhalten
- ◆ Reviewer werden darauf hingewiesen, dass sie ihrer Aufgabe nachkommen sollen

# *Programmerstellung*

- ◆ Aus der Menge der Bewertungen aller paper wird ein Programmvorschlag erstellt, basierend auf den Bewertungen der Reviewer
- ◆ Der Chair erhält eine Übersicht, die ihn über den Status jeden papers informiert
- ◆ Hier hat er die Möglichkeit, aufgrund dieser Einschätzung einzelne paper in das Konferenzprogramm aufzunehmen...
- ◆ ...oder abzulehnen
- ◆ ...oder den Reviewern wieder zu weiteren Bewertung vorzulegen
- ◆ **WICHTIG:** Der Chair entscheidet das Programm. Das Tool hat hier lediglich unterstützenden Charakter

# Forum

- ◆ Um die Diskussion eines Papers nicht auf wenige Zahlenwerte zu beschränken werden Diskussionsforen eingerichtet
- ◆ Hierbei gibt es drei Arten von Foren
  - ◆ Offene Foren
    - ◆ Hier kann jeder Beiträge erstellen, der an der Konferenz teilnimmt
  - ◆ Komiteeforen
    - ◆ Hier ist die Teilnahme auf die Reviewer und den Chair der Konferenz beschränkt
  - ◆ Paperspezifische Foren
    - ◆ Ausschließlich die Reviewer dieses Papers haben Zugang zu diesem Forum
- ◆ Offene und Komiteeforen werden vom Chair in beliebiger Anzahl eingerichtet
- ◆ Paperspezifische Foren werden automatisch zu jedem Paper erstellt
- ◆ Jedem Teilnehmer werden nur die Foren angezeigt, zu denen er Zugang hat

# *Zur Wahl von PHP*

- ◆ Insgesamt gerechnet ist PHP aus unserer Sicht die richtige Wahl für ein derartiges Projekt
- ◆ Es ist weit verbreitet, besitzt gute Zugriffsmöglichkeiten auf die Datenbank und ist leicht direkt in die Ausgabe zu integrieren
- ◆ Erschwerend sind hierbei jedoch manchmal auftretende Unterschiede zwischen Windows- und Linuxvarianten sowie teils recht grosse Unterschiede zwischen einzelnen Versionen
- ◆ Ausserdem wäre für ein grösseres Projekt ein besseres Errorhandling des Systems sehr wünschenswert

# Zur Gruppenarbeit

- ◆ Die Aufteilung der Module erwies sich als einigermaßen gut gewichtet
- ◆ Jeder wurde nach seinen Fähigkeiten mit entsprechenden Aufgaben betraut
- ◆ Für die Zeiteinteilung wäre ein früheres Setzen von verpflichtenden Milestones notwendig gewesen. Hier wurde viel versäumt.
- ◆ Regelmäßige Treffen **aller** Beteiligten wären notwendig gewesen
- ◆ Wären Gruppe und Projekt grösser gewesen, wäre das Projekt vermutlich nicht beendet worden, da zu wenig unternommen wurde, um die Gruppe als Ganzes zu koordinieren

# ***Gut dass es geschafft ist***

Gunnar Biederbeck  
Ivan Stragalis  
Marco Heyden  
Meiko Jensen  
Torben Dziuk  
Tim Fenten