

Mach-II

Ein objektorientiertes Framework

Präsentation bei der ColdFusion Usergroup
München, 19. Mai 2004

Harry Klein Software-Entwicklung

Übersicht

1. Hauptfragen von Entwicklern
2. CFC Grundlagen
3. Objektorientierung
4. Mach-II Grundlagen
5. Mach-II Bestandteile
6. Mach-II erweiterte Bestandteile
7. Weitere Informationen

1. Hauptfragen von Entwicklern

Was ist Mach-II?

- § objektorientiertes Framework
- § implizit aufgerufene Ereignisse

Wer hat Mach-II entwickelt?

- § Hal Helms (Fusebox, FLiP)
- § Ben Edwards (Fusebox J2EE, Adalon)

Mach-II Sprachen

- § ColdFusion
- § PHP
- § Java (in Entwicklung)

1. Hauptfragen von Entwicklern

Warum in ein neues Framework einlernen?

- § Vorteile der OOP, MVC Pattern
- § flexibel, modular und leicht erlernbar

Was ist mit Fusebox, CObjects, ...?

- § Fusebox - schnell erlernbares, flexibles Framework
prozedurale Programmierung
- § CObjects - durch die Einführung von CFCs überholt

Ist Mach-II langsamer als andere CF Frameworks?

- § wird komplett im Speicher gehalten
- § mindestens genauso schnell wie vergleichbare Frameworks

2. CFC Grundlagen

Mach-II besteht aus ColdFusion Components (CFCs):

- § wurden in CFMX eingeführt
- § sind der Einstieg in die Objektorientierung
- § ähneln Java Klassen
- § haben Einschränkungen: z.B. Konstruktor, Destruktor, Sicherheit, ...)

Beispiel: AppFactory.cfc

3. Objektorientierung

Momentan dominantes Programmierparadigma

Hauptbestandteile:

- § Objekte, bestehen aus:
 - § Eigenschaften
 - § Methoden
- § Kapselung
- § Vererbung
- § Polymorphismus

MVC Pattern (Model-View-Control)

- § Model: Datenmodell
- § View: Darstellungsschicht
- § Control: Steuerungsschicht

3. Objektorientierung - Kapselung

Gut gekapselte Objekte:

- § schützen ihre Daten vor Zugriffen von Außen
- § stellen Getter und Setter Methoden zur Verfügung
- § sind kohäsiv programmiert – d.h. haben einen einzigen Zweck

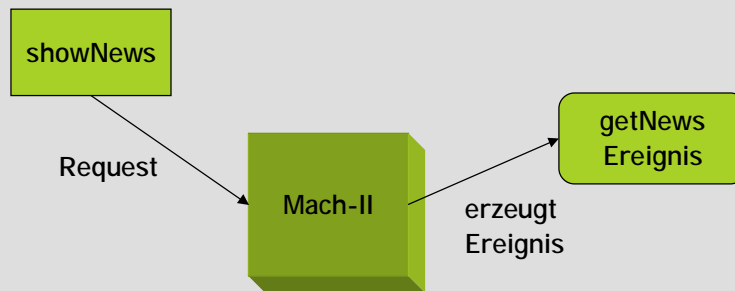
Beispiel User-Objekt:

- § Eigenschaft „isLoggedIn“ sollte geschützt werden
- § Methoden: `setIsLoggedIn()`, `getIsLoggedIn()`
- § Kohäsiv: In einem Userobjekt sollte z.B. keine Methode „printUserReport“ enthalten sein

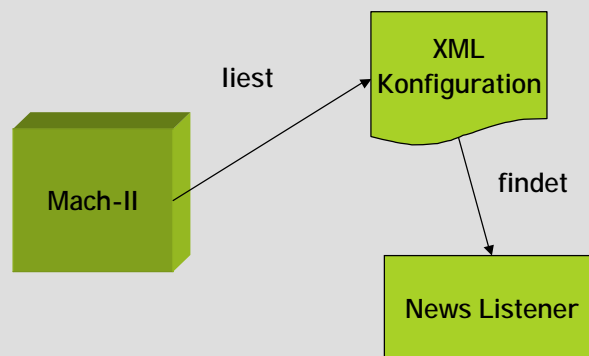
4. Mach-II Grundlagen

- § Mach-II ist ein objektorientiertes Framework
- § Unterstützt Entwickler bei der Erstellung eigener Applikationen
- § Architektur: *Implizit aufgerufene Ereignisse*
- § Was sind „implizit aufgerufene Ereignisse“?
 - § Anstatt Methoden direkt aufzurufen, werden Ereignisse angekündigt
 - § Ein Listener kann auf ein oder mehrere Ereignisse „reagieren“
 - § Das System bearbeitet ein Ereignis, indem es alle Methoden aufruft, die diesem Ereignis zugewiesen sind
 - § Ereignisse werden nacheinander, in einer Queue (Schlange) abgearbeitet

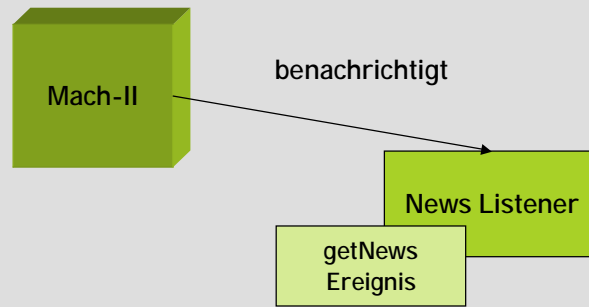
Beispiel: Implizit aufgerufene Ereignisse



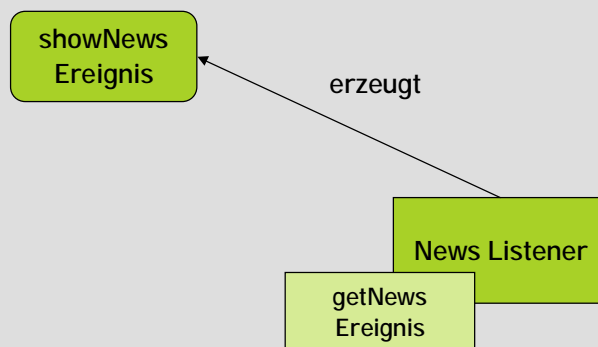
Beispiel: Implizit aufgerufene Ereignisse



Beispiel: Implizit aufgerufene Ereignisse



Beispiel: Implizit aufgerufene Ereignisse



5. Mach-II Bestandteile

- § Property
- § Listener
- § Event & Event Handler
- § View

Konfiguration: mach-II.xml

Beispiel: Contactmanager Konfiguration

Beispiel: MachWiki Konfiguration

Konfiguration: mach-II.cfm

Beispiel: Contactmanager Konfiguration

5. Mach-II Bestandteile - Property

```
- <properties>
  - <property name="defaultEvent" value="showHome" />
  - <property name="exceptionEvent" value="exception" />
  - <property name="applicationRoot" value="/ContactManager" />
  - <property name="eventParameter" value="event" />
  - <property name="parameterPrecedence" value="form" />
  - <property name="maxEvents" value="10" />
- </properties>
```

5. Mach-II Bestandteile - Listener

- § Schnittstelle zw. Framework und Business Modell
- § Enthalten Mach-II Business Logik
- § Werden vom Controller benachrichtigt; eine Methode des Listeners wird zu einem Ereignis aufgerufen
- § Können selbst Ereignisse erzeugen

5. Mach-II Bestandteile – Event & Event Handler

- § Kapseln der Request Information in ein “event” Objekt
- § Event Handler: Aufruf von Views, Benachrichtigen von Listener, Event Mappings, ...
- § Das aktuelle “event” is immer als request.event Variable definiert
- § Methoden: getName(), getArg(), setArg(), getArgs(), ...

6. Mach-II erweiterte Bestandteile - Filter

- § wiederverwendbare Framework Objekte, die Ereignisse und den Programmfluss manipulieren können
- § Werden in der Datei mach-ii.xml registriert
- § Unterschied zu Plugins: Filter werden nicht in jedem Request benötigt, sondern nur an einigen Stellen

Beispiel: RequiredFields Filter

```
<event-filter name="requiredFields"  
  type="MachII.filters.RequiredFieldsFilter" />
```

6. Mach-II erweiterte Bestandteile - Plugins

- § Werden in der Datei mach-ii.xml registriert
- § Mit Plugins kann die Funktionsweise der kompletten Applikation angepasst werden
- § Erweitern die Klasse "MachII.framework.Plugin"
- § Werden in der Datei "mach-ii.xml" registriert
- § Haben Einstiegspunkte: preProcess() , preEvent(), preView(), postView(), postEvent(), postProcess()

Beispiel: SimplePlugin

```
<plugin name="SimplePlugin"  
  type="MachII.plugins.SimplePlugin" />
```

7. Weitere Informationen

- § www.mach-II.com - offizielle Site von Hal Helms & Ben Edwards
- § www.corfield.org/index.php?event=machii.main - Sean Corfields Blog
- § www.mach-II.info - Phil Cruz FAQ
- § www.mach-II.de - deutsche Mach-II Site
- § Bei mir: klein@contens.de

Herzlichen Dank für Ihre Aufmerksamkeit



CONTENS Software GmbH
Oettingenstr. 25
80538 München
info@contens.de
www.contens.de