Software – Design
basierend
auf dem Plug-In – Konzept

Michael Antes

Seminar Simulation und Bildanalyse mit Java, WS2003

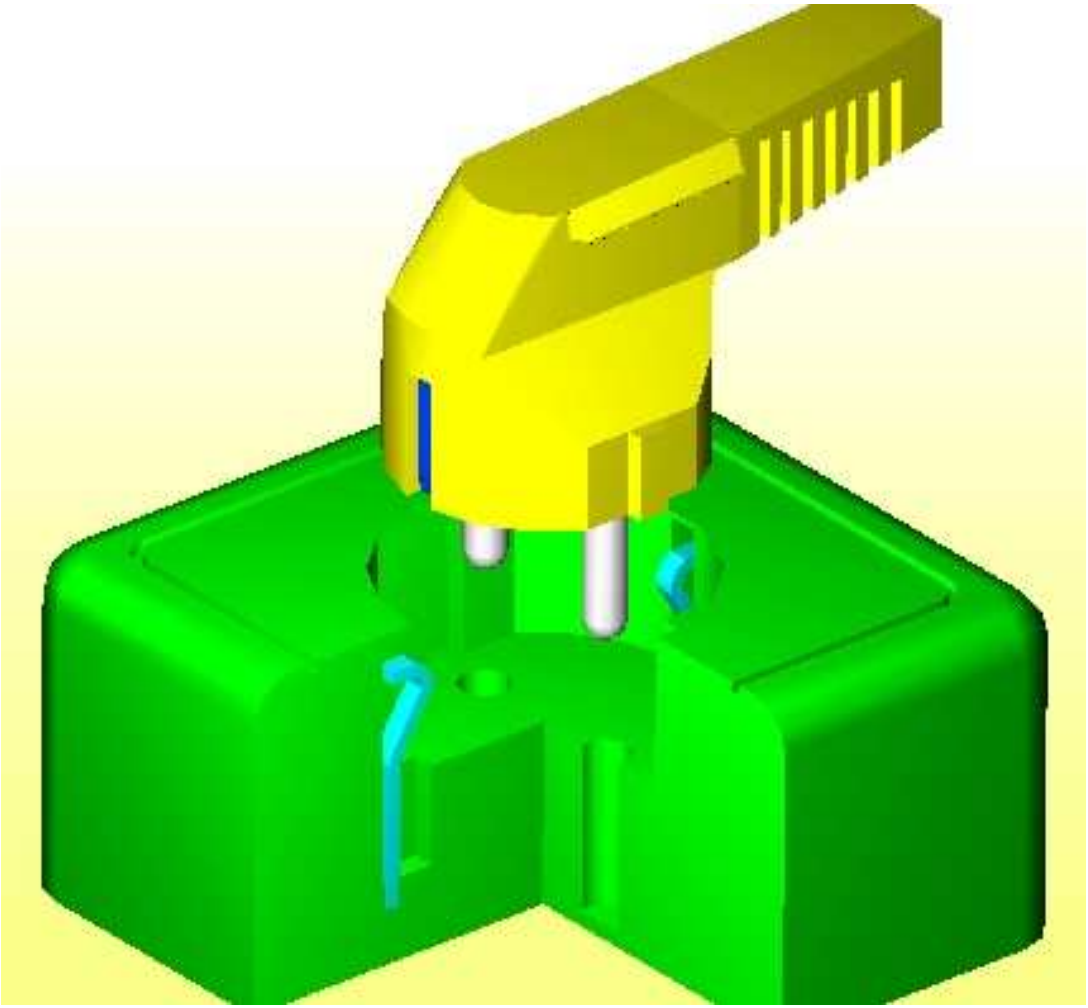
Universität Ulm

Software-Design basierend auf dem Plug-In-Konzept

Inhalt:

- Einführung: Was ist ein Plug-In?
- Plug-In-Entwurfsmuster
- Plug-In-Framework
- Literatur

Was ist ein Plug-In?



Was ist ein Plug-In?

Eigenschaften:

- Werden zu größeren Programmen hinzugefügt (neue / erweiterte Funktionalität)
- Bedienen bestimmte Schnittstellen
- Werden vollständig dynamisch geladen
- Nicht selbstständig lauffähig
- Können alleine verbreitet werden

Was ist ein Plug-In?
Warum Plug-Ins?

Motivierendes Beispiel: Grafikprogramm

- Programmierer kennt nicht alle Import-/Export-Algorithmen für alle Bildformate
- Neue Bearbeitungstechniken (Filter u.Ä.) sind einfach hinzuzufügen
- Programm startet langsam
- Programm benötigt viel Systemressourcen

Plug-In-Entwurfsmuster

Übersicht:

- Warum Entwurfsmuster?
- Anwendungsgebiete für Plug-In-Entwurfsmuster
- Schema des Plug-In-Entwurfsmusters

Plug-In-Entwurfsmuster

Warum Entwurfsmuster?

- Wiederverwendung von Lösungen
- Softwareentwürfe werden verständlicher
- Hilfestellung für unerfahrenere Programmierer

Plug-In-Entwurfsmuster

Anwendungsgebiete für Plug-In-Entwürfe (1):

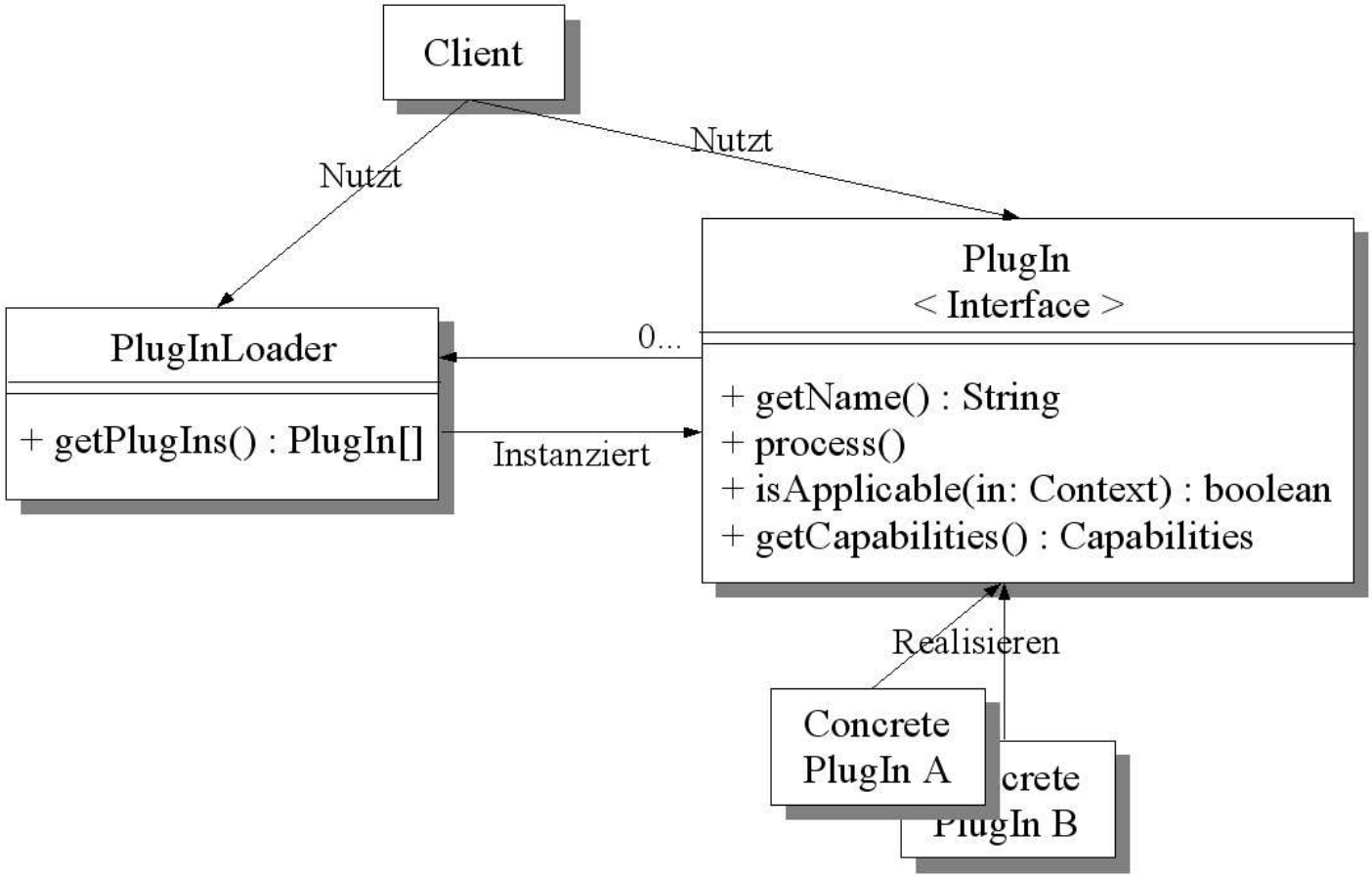
- Modularisierung großer Projekte
- Softwareentwicklung durch Dritte
- Erweiterungsbedarf während der Laufzeit
- Einfache Anpassung / Erweiterung nach Auslieferung

Plug-In-Entwurfsmuster

Anwendungsgebiete für Plug-In-Entwürfe (2):

- Flexibilität für Server
- Unabhängige Komponentenentwicklung
- Verringerung von Startzeiten und Hardwareanforderungen

Plug-In-Entwurfsmuster
Schema (UML)



Plug-In-Entwurfsmuster
Schema – Erläuterung (1)

- Client
  - Erhält Plug-Ins durch PlugInLoader
  - Entscheidet über Einsatz der Plug-Ins
  - Ruft passendes Plug-In auf
  - Kommunikation mit Plug-Ins einzig über das Interface

Plug-In-Entwurfsmuster
Schema – Erläuterung (2)

- PlugInLoader
  - Sucht dynamisch nach konkreten Plug-Ins
  - Liefert geladene Plug-Ins zurück

- ConcretePlugIn
  - Implementiert die PlugIn-Schnittstelle
  - Erbt evtl. von anderen Klassen, implementiert evtl. mehrere Schnittstellen

Plug-In-Entwurfsmuster
Schema – Erläuterung (3)

- PlugIn
  - Schnittstellendefinition
  - Methoden zur Ausführung (process())
  - Methode zur Namensrückgabe (getName())
  - „voting methods“ (isApplicable(Context c))
  - Beschreibung der Eigenschaften

Plug-In-Entwurfsmuster
Schema – Erläuterung (4)

- Plug-Ins gehen von einer Eignungsprüfung durch den Client aus
- „geeignete“ Plug-Ins können ihre Aufgaben auch erfüllen
- Der Client führt keine weitere Typprüfung mehr durch

Plug-In-Entwurfsmuster

Zusammenfassung:

- Plug-Ins erleichtern das Hinzufügen neuer Komponenten
- Vereinfachung des Projektdesigns durch Plug-In-Entwurf
- Nicht sämtliche Komponenten eines Projekts können als Plug-In realisiert werden
- Unabhängige Entwicklung der Plug-Ins

Plug-In-Entwurfsmuster
Beispielanwendungen

Beispiele:

- GimP
- SLC
- Netscape
- Adobe Photoshop

Plug-In-Framework in Java

Bestandteile:

- Vereinbarung von „Marker“-Interfaces
- Realisierung des PlugInLoaders

=> Bereitstellung der grundsätzlichen Funktionalität des Plug-In-Entwurfsmusters

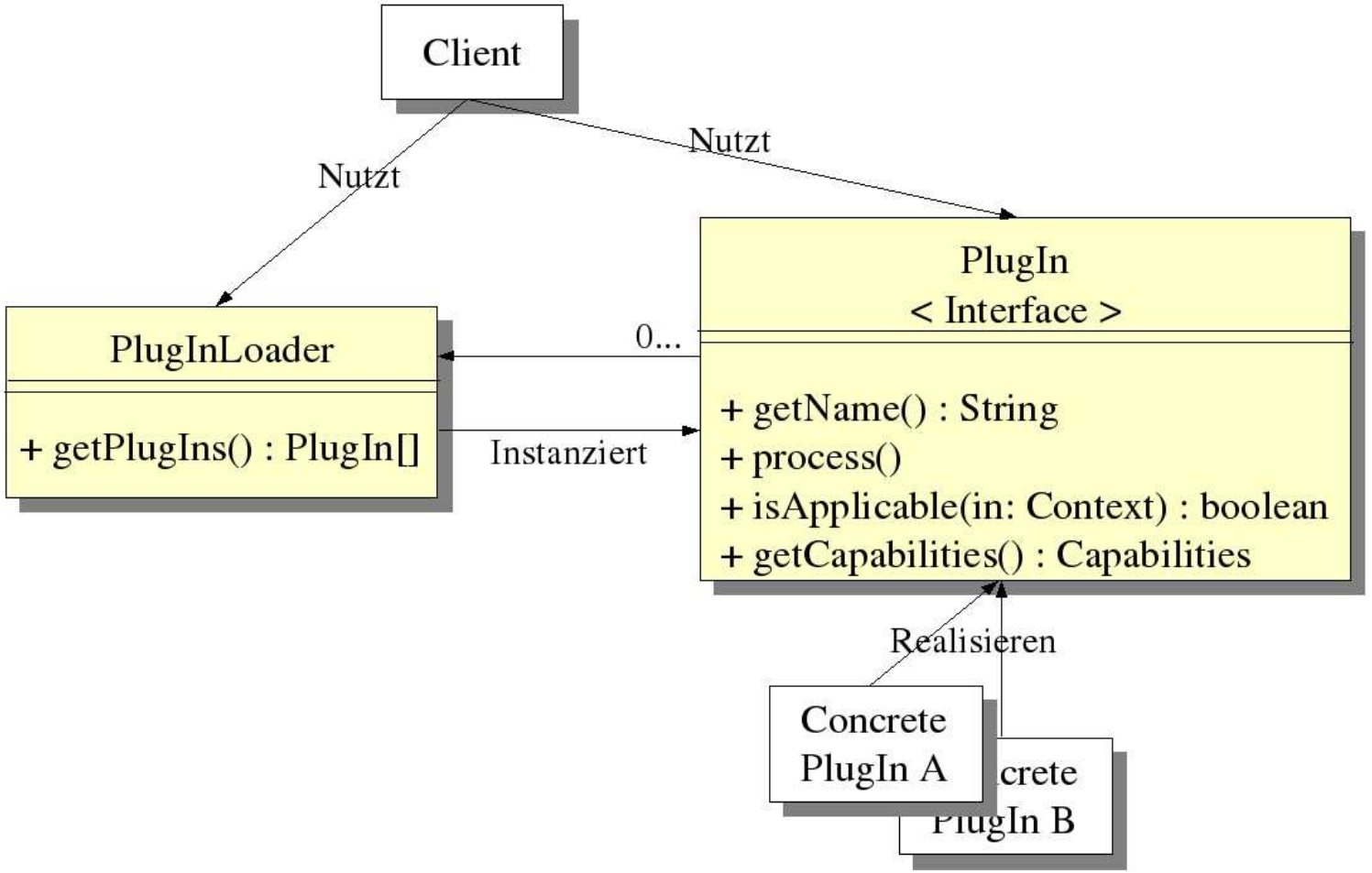
Plug-In-Framework in Java

Marker-Interface „PlugIn“:

```
public interface PlugIn {
}
```

Von diesem werden alle speziellen Plug-In-Interfaces abgeleitet

Plug-In-Framework in Java



Plug-In-Framework in Java

PlugInLoader:

- Realisierung der Suche nach Plug-Ins durch Dateisystemsuche.
- Instanziert passende, gefundene Plug-Ins
- Verwendbar für alle vom Markerinterface abgeleiteten Plug-Ins

Plug-In-Framework in Java

Verwendung des Frameworks:

- Evtl. Erweiterung der Schnittstellen
- Implementierung der Clients
- Implementierung der konkreten Plug-Ins als Realisierung der vereinbarten Schnittstellen

Literatur

Quellen:

- J. Mayer, On Quality Improvement of Scientific Software: Theory, Methods, and Application in the GeoStoch Development, Kapitel 2
- J. Mayer, I. Melzer, und F. Schweiggert, Lightweight Plug-in-Based Application Development