

Abschlußbericht Projektarbeit SVG/JavaScript Codegenerator für Multimediaanwendungen

Karin Leichtenstern
am 14.12.2004

Inhalt

- Kennzeichen einer Multimediaanwendung
- Das Metamodell
- Der Codegenerator
- Die Bibliothekselemente
- Zielsprache SVG
- Zielsprache JavaScript
- Zielcode nach MVC
- Fazit

Kennzeichen einer Multimediaanwendung

- Software mit mindestens einem kontinuierlichen (zeitabhängigen) Medium
- Hoher Grad an Benutzerinteraktionen und damit Bedarf an Ereignisbehandlungen
- Vielzahl an unterschiedlichen Benutzeroberflächen
- Komposition einer Vielzahl an Medien
- Vordefiniertes zeitliches Verhalten durch kontinuierliche Medien
- ⇒ Hohe Komplexität der Anwendung
- ⇒ Modellierung der Anwendung vor der Implementierung

3

Abschlußbericht
Karin Leichtenstern

Was ist ein Metamodell?

Metamodell ist ein Modell von einem Modell

Wie darf das Modell für MMA aussehen?
Berücksichtigung der Besonderheiten von MMAs
z.B. Regelung wie eine Klasse aussehen soll/darf.

- ⇒ Syntax und Semantik eines Modells
- ⇒ Regelwerk für MMAs

4

Abschlußbericht
Karin Leichtenstern

Vier Teile des Metamodells

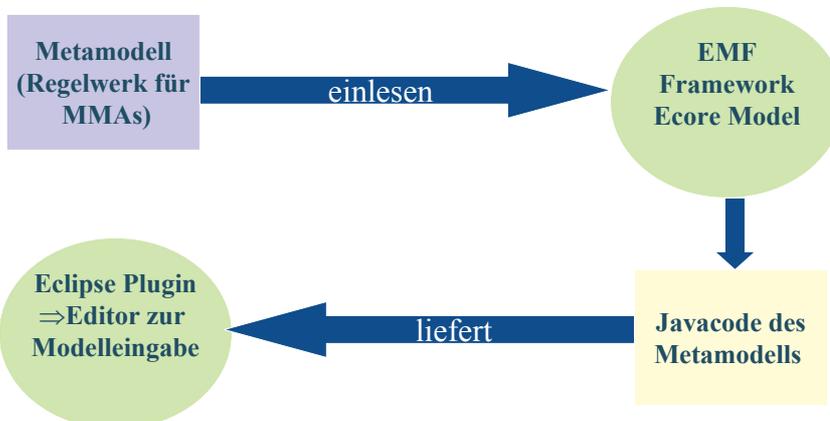
Metamodell definiert Regeln für:

- Klassendiagramm
- Zustandsdiagramm
- Präsentationsdiagramm
- Ereignisbehandlungsdiagramm

5

Abschlußbericht
Karin Leichtenstern

Der Codegenerator



6

Abschlußbericht
Karin Leichtenstern

Textbasierter Baumeditor



7

Abschlußbericht
Karin Leichtenstern

Der Codegenerator

Eclipse Plugin
⇒ Editor zur
Modelleingabe

Nutzereingabe

Konkretes
Modell einer
MM-Anwendung

Aufgabe der Projektarbeit

8

Abschlußbericht
Karin Leichtenstern

Aufgabenstellung

- Einarbeitung in die benötigten Technologien:
 - XMI (XML Metadata Interchange), EMF, JET
 - Modellierungssprache (Metamodell) für Multimedia-Anwendungen
- Untersuchung der Möglichkeiten von SVG/JavaScript
- Erarbeitung und Diskussion der Transformation nach SVG/JavaScript
- Aufzeigen von Problemen und ggf. Vorschlägen von Anpassungen der gegebenen Modellierungssprache
- Demonstration der Funktionsweise des Prototyps anhand ein oder mehrerer Beispiel-Anwendungen

9

Abschlußbericht
Karin Leichtenstern

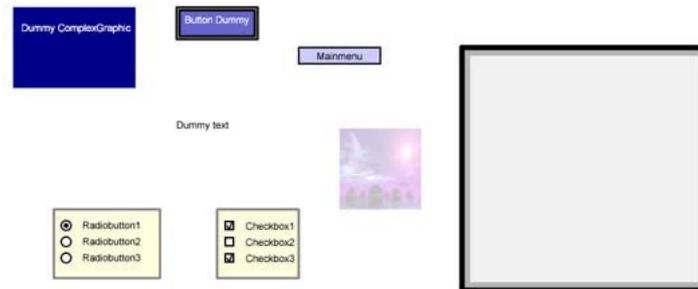
Bibliothek für User Interface Elemente des Codegenerators

Medien	Widgets
Komplexe Graphik	Button
Image	Textbox
Audio	Checkbox
Video	Radiobutton
Animation	Combobox
Text	

10

Abschlußbericht
Karin Leichtenstern

Bibliothek für User Interface Elemente des Codegenerators



11

Abschlussbericht
Karin Leichtenstern

Anforderung an die Zielsprache SVG

- Vektorbasierte Grafiksprache
- Unterstützung von kontinuierlichen Medien
- hoher Interaktionscharakter durch Events
- Dynamische Veränderung der Grafik

12

Abschlussbericht
Karin Leichtenstern

Anforderung an die Zielsprache SVG

- Vektorbasierte Grafiksprache ✓
- Unterstützung von kontinuierlichen Medien ✓
 - Animationskonstrukt
 - Video und Audio in Version 1.2
 - Einfache Einbindung externer Medien
- hoher Interaktionscharakter durch Events ✓
 - Aber keine Spezifikation zu Tastaturereignissen
- Dynamische Veränderung der Grafik ✓
 - DOM (Document Object Model)

13

Abschlußbericht
Karin Leichtenstern

Fazit zum Thema: SVG als Zielsprache von MMAs

- XML-basierte Markupsprache
 - Open Source Projekt
 - Erweiterung in der Spezifikation der Version SVG 1.2 ab Mai 2005 endgültige Spezifikation (laut W3C-Homepage)
 - Viele Vorteile für Anwendungen auf mobilen Endgeräten
 - geringer Speichergröße
 - eigene Spezifikation etc.
- ⇒ SVG 1.2 gut einsetzbar für Multimediaapplikationen

14

Abschlußbericht
Karin Leichtenstern

Anforderung an die Zielsprache JavaScript

- Objektorientierung
- Vererbung
- Datenkapselung
- Typisierung
- Ereignisbehandlung

15

Abschlußbericht
Karin Leichtenstern

Anforderung an die Zielsprache JavaScript

- Objektorientierung
 - Nur Objektbasiert
- Vererbung
 - Mit Hilfe des Prototypekonstruktes
- Datenkapselung
 - Keine Zugriffsreglung wie public, private, protected
- Typisierung
 - Schwache Typisierung
- Ereignisbehandlung
 - Veränderung des SVG Dokuments mittels DOM



16

Abschlußbericht
Karin Leichtenstern

Fazit zum Thema: JavaScript als Zielsprache von MMAs

- Grundsätzliche Arbeit mit JavaScript 1.5 möglich
 - Aber :
 - Prototypekonstrukt eher Notbehelf
 - Keine Interfaces für Mehrfachvererbung
 - Keine Datenkapselung und schwache Typisierung
 - JavaScript 2.0 sollen die Schwachstellen beheben
 - Klassenstruktur mit Vererbung
 - Datenkapselung, Packages etc. (ähnlich Java)
- ⇒ Für simple MMAs JavaScript 1.5 einsetzbar
⇒ JavaScript 2.0 wahrscheinlich sehr gut einsetzbar

17

Abschlußbericht
Karin Leichtenstern

Generierung des Codes nach MVC

DAS MODEL

Daten der Multimediaanwendung

Szenen

- Eigenschaften/Verhalten einer Szene
- Regelung von Szenenübergängen

Fachklassen/ApplicationEntities

- Eigenschaften/Verhalten von realen Objekten der MMA z.B. Car, Player

- ⇒ Informationen aus Klassendiagramm und Zustandsdiagramm
⇒ Pro Szene und Fachklasse eine JavaScriptdatei

18

Abschlußbericht
Karin Leichtenstern

Generierung des Codes nach MVC

DIE VIEW

Präsentation der Multimediaanwendung

Szenen

- Darstellung von Medien und Widgets
- Werfen Ereignisse
- Medien und Widgets repräsentieren Objekte vom Model z.B. ein Car

Application.svg & Application.js

- Container der Applikation.zum dynamischen Laden einer Szene
- Initialisierung der Objekte von Model & Control
- Steuerung des dynamischen Ladens

⇒ Informationen aus Präsentationsdiagramm

⇒ Pro Szene ein SVG Dokument

19

Abschlußbericht
Karin Leichtenstern

Generierung des Codes nach MVC

DAS CONTROL

Kontrolle der Multimediaanwendung

UI Elemente

1. Typische Eigenschaften/Verhalten eines UI Elementes (Widget & Medium)
2. Kontrollieren Benutzereingaben in der View
 - Ereignisbehandlung
 - Manipulation der View (DOM) und des Models

⇒ Informationen aus Klassen- und Ereignisbehandlungsdiagramm

⇒ Pro Repräsentation des Models durch ein UI Element in einer Szene
Generierung einer JavaScriptdatei (z.B. CarImageRace)

20

Abschlußbericht
Karin Leichtenstern

Fazit zum Thema: Metamodell für Multimediaanwendungen

- Gut durchdachtes Konzept
 - Berücksichtigung der Besonderheiten von MMAs
 - Im Besonderen:
 - Medieneinbindung/Repräsentation des Modells durch UI Elemente im Modell bereits integriert
 - Modellierung der Benutzeroberflächen möglich
 - Berücksichtigung des hohen Interaktionscharakters einer MMA
 - Verbesserungen im Ereignisbehandlungsdiagramm
 - UI Element eine typische Klasse?

21

Abschlußbericht
Karin Leichtenstern

Danke für die Aufmerksamkeit!

22

Abschlußbericht
Karin Leichtenstern