

## Übung 6 – Multimedia-Programmierung

### Inhalt

- Modellierung mit MML unter Verwendung von MagicDraw mit dem MML-Plugin
- Aufgabe 2 entspricht der HA 3

### Aufgaben

#### Starten von MagicDraw mit dem Plugin für MML:

- Um die MML-Diagramme verwenden zu können, muss MagicDraw mit `/home/proj/mi_mmp_ss07/MD_UML_116_PE/bin/mduml` gestartet werden. (Falls jemand auf dieses Verzeichnis keine Zugriffsrechte haben sollten, bitte melden)
- In den nachfolgenden Dialogen kann angegeben werden:
  - Importieren von Benutzerdaten: *Default*
  - Modus: Aktueller Benutzermodus + Experte (Checkbox unten!)
- Das Beispiel-Projekt, das beim Start angezeigt wird, kann einfach geschlossen werden und es kann ein neues Projekt im eigenen Home-Verzeichnis erstellt werden.

#### Aufgabe 1: Multimedialer Online-Shop

Es soll ein einfacher multimedialer Online-Shop für Handys entwickelt werden. Im Katalog des Händlers befinden sich Handys. Jedes Handy hat einen Hersteller, Bezeichner und einen Preis. Außerdem steht zu jedem Handy ein Dokument mit einer textuellen Beschreibung, ein Photo und ein 3D-Modell zur Verfügung.

Auf der Startseite des Shops kann der Benutzer ein Gerät auswählen. Dazu wählt er entweder aus den angezeigten Standard-Vorschlägen oder er gibt einen Suchbegriff (Herstellernamen und/oder Bezeichner) ein woraufhin in einer zweiten Auswahl seine Suchtreffer angezeigt werden. Bei allen Handys ist dabei stets das zugehörige Photo zu sehen.

Wird ein Handy ausgewählt, so gelangt man zur Detailansicht. Dort sind neben Hersteller und Bezeichner auch das Photo, die textuelle Beschreibung und die 3D-Animation des Handys sichtbar. Der Benutzer kann im 3D-Modell das Handy auf- oder zuklappen (der Einfachheit halber wird angenommen, dass alle Handys aufgeklappt werden können).

Der Benutzer kann anschließend entweder das Handy bestellen, oder zu seinen bisherigen Suchergebnissen zurückkehren. Im Falle einer Bestellung erhält der Benutzer eine Bestätigung (soll hier nicht weiter modelliert werden) und anschließend wird die Anwendung beendet. Weitere Details (wie z.B. Kundendaten) brauchen hier nicht spezifiziert werden.

Erstellen Sie ein MML-Modell zur Beschreibung der Anwendung. Erstellen Sie dazu ein *Structure Diagram*, ein *Scene Diagram*, zwei *User Interface Diagrams* für die Suche und die Detailanzeige und ein *Interaction Diagram* für die Detailanzeige.

Hinweise zum MML-Plugin in MagicDraw:

- Sub-Animation: Stereotyp muss von Hand eingegeben werden (siehe Vorlesung).
- Transitions: Im Scene Diagram werden bei den Transitionen die jeweils aufzurufende Entry-Operation (der nächsten Scene) aufgerufen. Dazu im Eigenschaftsfenster der Transition:
  - im Abschnitt *Trigger* als *Ereignistyp CallEvent* angeben,
  - anschließend bei *Operation* gewünschte Entry-Operation auswählen (muss bereits existieren).
  - Das Schlüsselwort *history* kann momentan nur als beschreibender Text (neues Textelement) daneben platziert werden.
- Entry-Operations: Bei Entry-Operations von Scenes muss manuell der Stereotyp *EntryOperation* hinzugefügt werden
- (Exit-Operations: werden meist nicht explizit modelliert sondern es wird angenommen, dass jede Exit-Operation automatisch den gleichen Namen erhält wie die zugehörige Transition mit *exitTo\_* als Prefix, also z.B. *exitTo\_startMenu()*)
- User Interface: Um von den User Interface Components die graphische Darstellung als Icon zu erhalten, müssen Methoden und Attribute ausgeblendet werden (kleines Minuszeichen im Klassensymbol)
- (Zum Schluss: Für automatische Codegenerierung müsste im Structure Diagram jede Scene mit dem zugehörigen Presentation Unit und Activity über eine Abhängigkeit verbunden werden)

**Aufgabe 2 ( = Hausaufgabe 3): Auto-Rennspiel**

Modellieren Sie ein kleines Autorennspiel mit MML. Es gibt einen Spieler, das Fahrzeug, verschiedene Strecken sowie Hindernisse auf den Strecken. Jeder Spieler hat einen Namen. Das Fahrzeug hat x- und y-Position, eine Ausrichtung und eine Geschwindigkeit. Weiterhin hat es die Methoden *steerRight()*, *steerLeft()*, *accelerate()*, *brake()* und *drive()*.

Es soll 3 Szenen geben: *Menu*, *Game* und *Help*. Im Menü wird ein Spieler erstellt und eine der abgebildeten Strecken ausgewählt und anschließend das Spiel gestartet. Alternativ kann das Spiel dort auch beendet werden.

Im Spiel werden der Name des Spielers angezeigt, sowie Fahrzeug, Strecke und Hindernisse mittels Animationen dargestellt. Der Spieler kann durch vier verschiedene UI-Componenten die Methoden *steerRight()*, *steerLeft()*, *accelerate()* und *brake()* auslösen. Das Fahrzeug wird in regelmäßigen Abständen (Time-Sensor) durch Aufruf der Methode *drive()* fortbewegt. Fährt das Fahrzeug durch eine Kurve (*steerLeft()* oder *steerRight()*) so werden die Räder des Fahrzeugs entsprechend gedreht (Sub-Animation sowie eine entsprechende Methode der Medienkomponente). Prallt das Fahrzeug auf ein Hindernis, wird es abgebremst (Aufruf der Methode *brake()*). Berührt das Fahrzeug das Ziel (Ziel ist ein bestimmter Bereich innerhalb der Strecke), so ist das Spiel beendet und der Benutzer kommt zurück ins Menü. Von jedem Punkt der Anwendung aus kann die Hilfe aufgerufen werden.

**Abgabe:**

- Einzureichen ist eine MagicDraw-Projektdatei (\*.mdzip), mit folgenden Diagrammen **für Aufgabe 2:**
  - Structure Diagram,
  - Scene Diagram,

- User Interface für die Szene *Game*
- Interaction Diagram für die Szene *Game*
- (d.h. für *Menu* und *Help* brauchen keine User Interface Diagramme und Interaction Diagramme erstellt werden).
- Benennen Sie die Datei mit ***Vorname.Nachname***
- Es werden nur individuelle Lösungen angenommen
- Schicken Sie Ihre Lösung bis spätestens **7. Juni 2007** per Email an **andreas.pleuss@ifi.lmu.de**