

Metamodellierung

Entwicklung der Modellierungssprache Simple Web Modeling Language (SWML) für Web Anwendungen.

Ausgangsszenario

Eine Softwareentwicklungsfirma entwickelt wiederholt einfache Web-Anwendungen, die alle durch die gleiche Menge an Konzepten beschrieben werden können. Es soll eine Modellierungssprache für solche Web-Anwendungen entwickelt werden, welche in weiterer Folge automatisierte Code-Generierung ermöglichen soll (siehe zukünftige Übungsbeispiele).

Die Konzepte von SWML werden ausgehend von einem Beispiel-Modell erklärt. Wie in Abbildung 1 erkennbar, umfassen die Konzepte der Modellierungssprache unter anderem:

- Entity Typen (zB. Lehrpersonal, LVA, und Pruefung) und Relationships für die Modellierung der Daten in der Web-Anwendung,
- Web Pages (zB. Institut, Personal-Index, LVA, PruefungErstellen), sowie Links (verschiedene Pfeile) für die Modellierung des Web Interfaces.

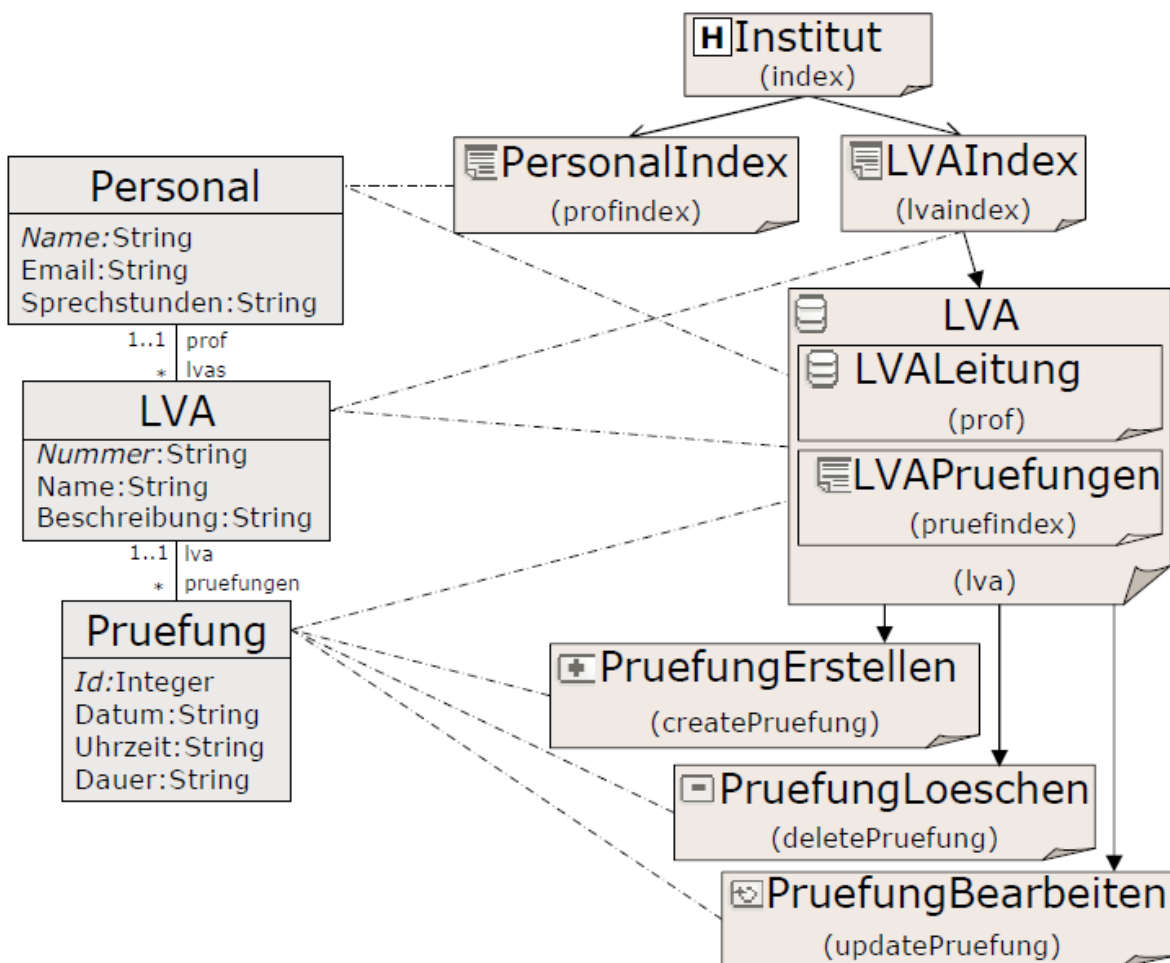











Abbildung 1: Beispiel Modell

Die Syntax und Semantik der Modellelemente wird wie folgt definiert:

Syntax		Semantik
<div>Personal</div> <div>Name:String Email:String Sprechstunden:String</div>		Entities haben einen eindeutigen Namen, sowie eine Menge von Attributen. Entities haben ein Attribut, das als ID verwendet wird.
	Das ID-Attribut wird kursiv hervorgehoben.	Attributes haben ebenfalls einen Namen und einen Typ. Erlaubte Typen sind String, Integer, Float, und Boolean.
	Ist die lower Multiplizität gleich 0, kann sie weggelassen werden.	Relationships repräsentieren semantische Beziehungen zwischen zwei Entity Typen. An beiden Relationship-Enden werden Multiplizitäten und Rollennamen notiert. Jedes Ende hat damit einen rollenamen, eine lower Multiplizität und eine upper Multiplizität. Das Konzept der Relationships kann beispielsweise ähnlich dem Reference-Konzept in Ecore modelliert werden. Multiplizitäten werden oftmals als Typ Integer modelliert. „*“ ist dann gleichbedeutend zu „-1“
	Die Startseite wird durch ein Icon („H“) gekennzeichnet.	Web Pages haben einen Namen/Titel und eine eindeutige relative URL (zB. die Web Page „Institut“ hat die relative URL „index“). Genau eine Web Page (dh. genau genommen eine Static Page) ist die Startseite der Web-Application. Diese Web Page ist später von jeder anderen Web Page der Web-Anwendung navigierbar. Web Pages treten in gewissen Spezialisierungen auf:
		Static Pages präsentieren beliebigen statischen, aber keinen dynamischen Inhalt aus dem Datenmodell. Zum Beispiel könnte eine Static Page eine Sammlung nützlicher Links darstellen.
	Dynamic Pages haben alle spezielle Icons. Die Beziehung zur Entity wird durch eine gestrichelte Linie dargestellt.	Dynamic Pages hingegen präsentieren zur Laufzeit generierten Inhalt aus dem Datenmodell. Damit hat eine Dynamic Page immer eine Beziehung zu dem Entity Typ dessen Instanzen (die Entities) angezeigt werden sollen. Dynamic Pages werden unterschieden in Entity Pages und Index Pages.
		Index Pages zeigen mehrere Entities eines Entity Typen in Form einer (verlinkten) Liste an, zB. Eine Liste aller angebotenen Lehrveranstaltungen eines Instituts.

 Personal (prof)		Entity Pages zeigen immer eine Entity an, zB. Informationen zu einer konkreten Lehrperson. Die Entity Page kann weiter spezialisiert werden in Create Page, Update Page, und Delete Page.
 Pruefung Erstellen (createPruefung)		Create Page erlaubt eine Entity zu erzeugen und zeigt dafür ein leeres Formular für den bestimmten Entity Typ an. Ist das Erzeugen erfolgreich, dann soll der User zur Ausgangseite (in unserem Fall LVA) umgeleitet werden.
 Pruefung Bearbeiten (updatePruefung)		Update Page erlaubt eine Entity zu bearbeiten und zeigt dafür ein mit den aktuellen Werten befülltes Formular an. Ist das Bearbeiten erfolgreich, dann soll der User zur Ausgangseite (in unserem Fall LVA) umgeleitet werden.
 Pruefung Loeschen (deletePruefung)		Delete Page erlaubt eine Entity zu löschen, zeigt dafür noch einmal die Informationen zur Entity an und bietet eine Möglichkeit zum Löschen (Button). Ist das Löschen erfolgreich, dann soll der User zur Ausgangseite (in unserem Fall LVA) umgeleitet werden.
<div>  LVA </div> <div>  LVALeitung (prof) </div> <div>  LVAPruefungen (pruefindex) (lva) </div>		Inner Pages. Dynamic Pages können innerhalb von Entity Pages geschachtelt werden. Die Inner Pages zeigen Informationen abhängig von der Entity Page, zB. zeigt die Inner Page LVALeitung Informationen zur Lehrperson die, die konkrete Lehrveranstaltung (Entity Page LVA) leitet. Die relativen URLs der Inner Pages beziehen sich ebenfalls auf die Entity Page, dh. sie sind als Anker im Sinne von HTML zu verstehen.
	Verschiedene Pfeilarten.	Links stellen die Navigationsmöglichkeit zwischen Web Pages dar. Links haben immer eine Source Web Page und eine Target Web Page. Auf jener Web Page (Source), von der der Pfeil wegführt, wird ein Link auf die Target Web Page angezeigt. Damit kennt eine Web Page üblicherweise die Links für die sie als Source, aber nicht jene, für die sie als Target auftritt. Einem Link kann ein Name zugeordnet sein. Man unterscheidet zwei Linkarten:
		Non Contextual Links sind gewöhnliche Links, dh. sie transportieren keine Information.

		Contextual Links transportieren Informationen zur Target Web Page, die diese für die Präsentation ihrer Inhalte benötigt, zB. um Informationen zu einer konkreten Lehrveranstaltung anzeigen zu können.
---	--	---

Aufgabenstellung

Entwickeln Sie mit EMF ein Metamodell basierend auf Ecore in UML Klassendiagramm- Notation für SWML. Achten Sie dabei darauf, dass das Metamodell möglichst präzise ist, d.h., das Metamodell soll alle notwendigen Sprachkonzepte durch entsprechende Klassen, Attributen, Beziehungen, Rollennamen usw. erfassen, und die Zusammenhänge zwischen den Konzepten möglichst genau einschränken. Beachten Sie dabei, dass nicht jedes Konzept (zB. aufgrund von Vererbung) auch eine graphische Repräsentation besitzt. Verwenden Sie soweit möglich die in der Angabe verwendeten Begriffe, und führen Sie eigene Begriffe nur ein, wenn aus der Angabe keine geeigneten zu entnehmen sind. Geben Sie für eigene Begriffe Definitionen von Syntax und Semantik an auf Basis obiger Angaben.

Implementieren Sie das Metamodell mittels EMF. Nutzen Sie die Funktionalität von EMF, um einen Baum-basierten Editor zu erstellen, und testen Sie Ihr Metamodell, indem Sie das oben angegebene Beispielmmodell mit diesem Editor modellieren.