



Nicolas
Karrer

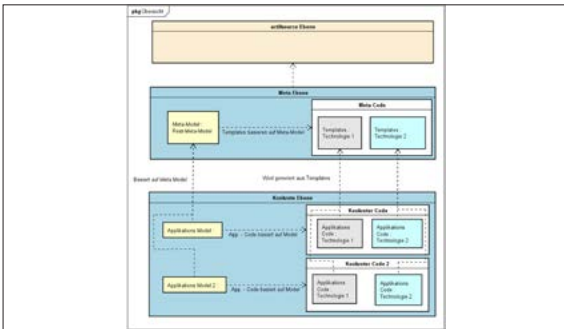


Marco
Sonderegger

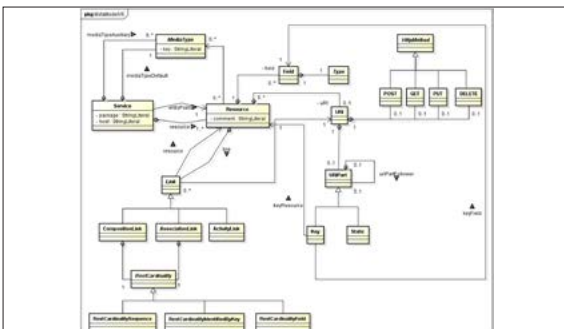
Diplomanden	Nicolas Karrer, Marco Sonderegger
Examinator	Prof. Dr. Olaf Zimmermann
Experte	Dr. Gerald Reif, ipt, Zug
Themengebiet	Software
Projektpartner	actifsource GmbH, Baden-Dättwil, AG

REST Hypermedia Modelling & Visualisation

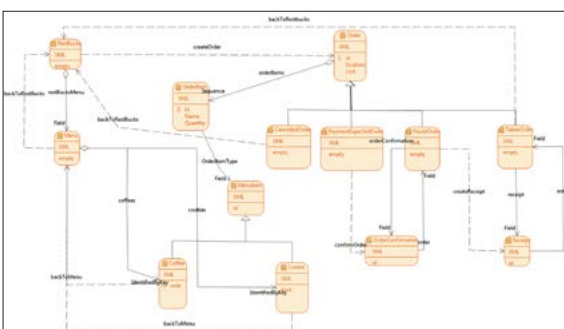
RESTful Applikationsentwicklung auf REST Maturitätslevel 3



Übersicht über die Architektur von REST Hypermedia Modelling & Visualisation



REST-Meta-Modell



Modellierte REST-Schnittstelle mittels eines eigenen Diagrammtypen in der actifsource-Umgebung

Ausgangslage: Webapplikations-Schnittstellen sollen die Prinzipien des Representational State Transfer (REST) erfüllen. Die REST-Prinzipien sind:

- REST-Schnittstellen stellen Ressourcen im Netzwerk zur Verfügung.
- Ressourcen unterstützen unterschiedliche Repräsentationen.
- Die Kommunikation zwischen Client und Server ist zustandslos.
- Der Zugriff auf die Ressourcen erfolgt über HTTP-Methoden.
- Die REST-Schnittstellen veranlassen, dass mit Hyperlinks die Aktionen, welche ein Client ausführen darf, diesem bekannt gemacht werden.

Der Client ist somit vom Server entkoppelt. Das Ziel der Bachelorarbeit ist es, REST-Schnittstellen zu modellieren. Es soll den Entwicklungsaufwand für den Programmcode reduzieren. Dazu wird ein schnittstellenspezifischer Code erzeugt.

Vorgehen/Technologien: Um dieses Ziel zu erreichen, wählen wir ein agiles Vorgehensmodell. Wir teilten die Durchläufe in vier Schritte auf:

- Analysieren der Problemstellung sowie REST-Thematik und Erstellen von repräsentativen Beispielen,
- Entwerfen des Meta-Modells,
- Entwickeln des Meta-Codes zur Generierung spezifischer Codes,
- Implementieren und Testen des Verhaltens sowie der Vergleich der REST-Schnittstellen mit den Kriterien der Ausgangslage.

Ergebnis: Die beiden Hauptergebnisse der Arbeit sind ein REST-Meta-Modell und ein Meta-Code zur Generierung eines spezifischen Codes für REST-Schnittstellen. Damit wird es möglich, in der Eclipse-Erweiterung actifsource-REST-Schnittstellen zu modellieren und deren Code zu generieren. Die Lösung fokussiert auf:

- grafischem Modellieren der REST-Ressourcen und deren Verbindungen,
- dem Konfigurieren der REST-Schnittstelle,
- dem Generieren von Code.

Für die Modellierung der REST-Schnittstellen führten wir einen neuen actifsource-Diagramm-Typ ein. Entwickler benutzen diesen, um REST-Schnittstellen zu erstellen. Die Konfiguration der REST-Schnittstelle bezieht sich auf die Definition der Medientypen, die Attribute und die Adressierung der Ressourcen mittels konfigurierbaren URIs. Die Applikation generiert zwei verschiedene Arten von technologiespezifischem Code. Zur Vollständigkeit ist noch das manuelle Einbinden der Businesslogik nötig. Der Code aller REST-Schnittstellen ist einheitlich und übersichtlich. Das Ziel der Aufwandsreduktion wurde erreicht, denn der Entwickler muss nur noch 10–20% des Codes selbst schreiben.