

KOLLOQUIUM

Mathias Slawik, HTW Berlin

31.10.2011

Gliederung

2

- Masterarbeit - Zusammenfassung
- Darstellung GenerationREST
- Fazit und Ausblick

3

Masterarbeit - Zusammenfassung

Projekt Developer Garden

4

- Bereitstellung von Telekom-Infrastruktur als Webservices (REST/SOAP)
 - ▣ Telefongespräche und –konferenzen, SMS & MMS, etc.
- Unterstützung durch Internet-Portal
 - ▣ Blog, Forum, etc.
- Bereitstellung
 - ▣ Benutzerdokumentation (REST/SOAP – Beschreibung)
 - ▣ SDKs für Java, PHP & .NET

Ausgangssituation

5

- Seit Jan 2010: Modellgetriebene Entwicklung im Developer Garden
 - ▣ Entwicklung eines Frameworks zur Generierung von Benutzerdokumentation und REST SDKs
 - Erfolge
 - ▣ Automatisierung der Pflege der Benutzerdokumentation
 - ▣ Ablösung der vorhandenen extern entwickelten SDKs
 - ▣ Qualitätssteigerung, Kostenreduktion
- ⇒ Entstehung diverser Herausforderungen

Herausforderungen und Lösungsansätze (1)

6

- Grundlegende Änderung der Anforderungen, aber keine grundlegende Änderung der Architektur
 - ▣ Komplexe Implementierung, „Code-Smell“
 - ▣ keine Abstraktion generischer REST-Eigenschaften aus Telekom-Implementierung

Lösungsansätze:

- Entwurf einer neuen, flexiblen Architektur für derzeitige und kommende Anforderungen
 - Abstraktion über REST – Architekturprinzip anstatt Telekom-Spezifika
- ⇒ Erweiterung des Anwendungsbereichs, strukturiertere Implementierung

Herausforderungen und Lösungsansätze (2)

7

- Risiko: (mittlerweile) veraltete Technologien

Lösungsansatz:

Implementierung auf Basis von Eclipse Indigo
(Release 22.06.2011)

- Pflege des Service-Modells über strukturierten Editor
 - ▣ Relativ aufwändig, nicht selbsterklärend
 - ▣ Keine Erzwingung semantischer Korrektheit
 - ▣ Basis: XML

Lösungsansatz:

Definition einer intuitiven textuellen DSL

Herausforderungen und Lösungsansätze (3)

8

- Modelltransformation durch interpretierte Sprachen
 - ▣ Erhebliche Performanceprobleme
- IDE-Integration suboptimal
 - ▣ S. Efftinge, Project Lead:
„The tooling is hand-made with JFace and Co, the code is old, not well tested and unfortunately a bit buggy.”

Lösungsansatz:

Verwendung von Xtext 2.0 und Xbase

Herausforderungen und Lösungsansätze (4)

9

- Keine Modellierung der Zielsprachen
 - ▣ Komplexe Code-Templates

Lösungsansatz:

Objektorientierte Strukturierung der Code-Templates

Herangehensweise

10

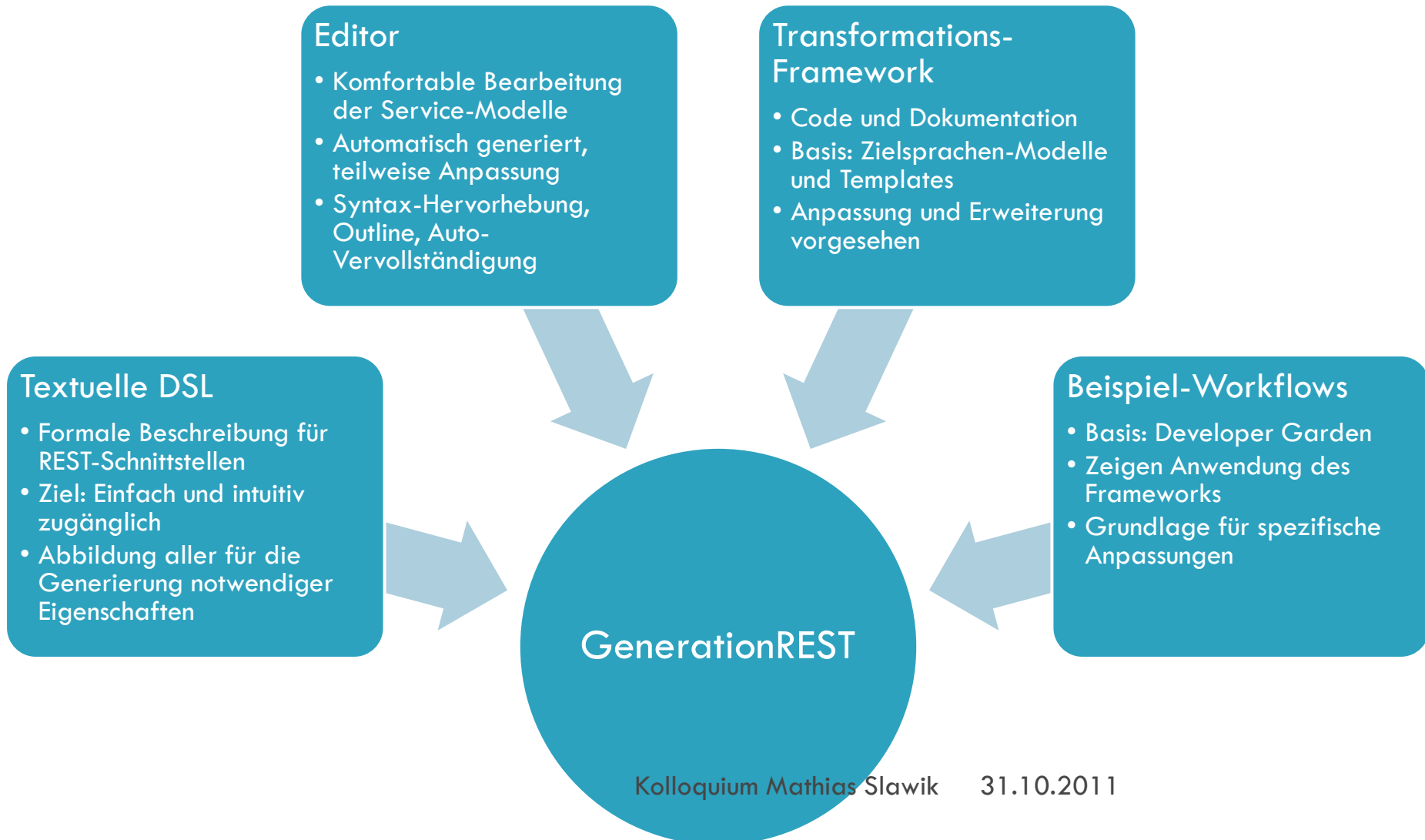
1. Umfassende Technologieanalyse
2. Analyse des vorhandenen Frameworks
3. Entwurf GenerationREST-Architektur
4. Implementierung des Prototyps
5. Einordnung, Fazit, Ausblick

11

Darstellung GenerationREST

Bestandteile GenerationREST

12



Modellpflege über textuelle DSLs

13

```
Field -> DocumentedElement
  DocumentedElement
  optional : EBoolean
  multiple : EBoolean
ValueField<T extends DataType> -> Field
  T extends DataType
  Field
  dataType : T
  staticValue : EString
StructuredField<F extends Field> -> Field
  F extends Field
  Field
  fields : F
```

```
GET /location => Method locateIP : DGMethod {
  parameters {
    DGString ipAddress
  }
  response {
    results[] {
      DGString ipAddress
      DGString radius
      isInRegion? {
        DGString countryCode
        DGString region
        DGString regionName
      }
    }
  }
}
```

```
Rest Method locateIP
  Rest Value Field ipaddress
  Rest Structured Field results
    Rest Value Field ipAddress
    Rest Value Field radius
    Rest Structured Field isInRegion
      Rest Value Field countryCode
      Rest Value Field region
      Rest Value Field regionName
  Rest Resource Path Static Element location
  Rest Resource Path Static Element p3gw-mod-odg-ipl
  Rest Resource Path Static Element rest
  Rest Resource Path Dynamic Element environment
```

```
/*
 * Fields
 */
RestValueField returns rest::RestValueField:
  dataType=[rest::RestDataType] (multiple?="[ ]"? name=EString (optional?="?")? (staticValue=STRING)?
;

RestStructuredField returns rest::RestStructuredField:
  name=ID (multiple?="[ ]"? (optional?="?")? '{'
  fields+=(RestValueField|RestStructuredField)+
  '}'
;
```

Modellierung der Zielsprachen

14

```
JavaModule -> File, NamedElement, VisibleElement, ModifiableElement
  File
  NamedElement
  VisibleElement
  ModifiableElement
  interface : EBoolean
  package : EString
  inherits : JavaModule
  methods : JavaMethod
  fields : JavaField
ServiceInterface -> JavaModule
  JavaModule
  restService : RestService
ServiceImplementation -> JavaModule
  JavaModule
  serviceInterface : ServiceInterface
```

```
class ServiceInterfaceExtension {
  def getPackage(ServiceInterface si) {
    si.restService.rootPackageName + ".service"
  }

  def getName(ServiceInterface si) {
    "I" + si.restService.name.toFirstUpper + "Service"
  }

  def isInterface(ServiceInterface si) {
    true
  }
}
```

```
class ServiceImplementationExtension {
  def getPackage(ServiceImplementation impl) {
    impl.serviceInterface.^package + ".impl"
  }

  def isInterface(ServiceImplementation impl) {
    false
  }

  def EList<JavaModule> create inherits: new BasicEList<JavaModule>() getInherits(ServiceImplementation impl)
  {
    inherits.add(impl.serviceInterface)
  }
}
```

Dokumentation der Modellelemente

15

4.7.26. REST Daten-Struktur **ParticipantStatus** - Informationen zum Status eines Teilnehmers

Tabelle 4.121. Felder des Typs **ParticipantStatus**

Feld	Beschreibung
number	Die gewählte Telefonnummer.
last_reason	Grund eines Verbindungsabbruchs.
status	Aktueller Verbindungsstatus. Mögliche Werte sind: <ul style="list-style-type: none">• Predial• Connecting• Connected• Joined• Finished• Removed

conferenceCall	ParticipantStatus	number	Die gewählte Telefonnummer.
conferenceCall	ParticipantStatus	status	Aktueller Verbindungsstatus. Mögliche Werte sind: <ul style="list-style-type: none">• Predial• Connecting• Connected• Joined• Finished• Removed

```
/p3gw-mod-odg-ccs/rest/{Environment : environment} => Service conferenceCall {  
  types {  
    ParticipantStatus {  
      DGString number  
      DGString last_reason  
      DGString status  
      DGString muted  
      DGString last_access_time  
    }  
  }  
}
```

```
* @return  
* <p>  
* Aktueller Verbindungsstatus.  
* <p>  
* Mögliche werte sind:  
* <ul>  
* <li>{@code Predial}  
* <li>{@code Connecting}  
* <li>{@code Connected}  
* <li>{@code Joined}  
* <li>{@code Finished}  
* <li>{@code Removed}  
* </ul>  
*  
*/  
public String getStatus() {  
  return status;  
}
```

Slawik 31.10.

GenerationREST

16

Innovativ

- Kein vergleichbares Framework

Aktuell

- Neueste Eclipse-Technologien

Universell

- Verwendbar für (fast) alle REST-Webservices

Offen

- Flexibles Open Source Produkt

Strukturiert

- Erkenntnisse aus den letzten 2 Jahren eingearbeitet

Ausblick

17

- Geplante Fertigstellung bis Mitte Dezember
 - ▣ Komplette Framework-Funktionalität migriert
 - Veröffentlichung des Frameworks
 - ▣ Marketing durch Developer Garden
 - ▣ TechTalk in der c-base, voraussichtlich am 16.02.2012
 - ▣ Bereitstellung über www.generationrest.org
 - Unklar: Nutzung durch Telekom
 - ▣ Probleme: fehlendes Know-How, Ressourcenknappheit
- ⇒ Ziel: Adaption durch weitere Projekte

18

Ende