



Ein Komponenten-Framework für die situationsabhängige Adaption Web-Service-basierter Standardsoftware

Michael Amberg, Shota Okujava,
Jens Wehrmann

Lehrstuhl für Wirtschaftsinformatik III
Friedrich-Alexander-Universität
Erlangen-Nürnberg



Agenda

- **Motivation**, Einführung und Einordnung
- **Situationsabhängige Adaption**
 - **Ziele** situationsabhängiger Adaption
 - **Szenarien** situationsabhängiger Adaption
- **Komponenten-Framework**
 - **Konzeption** und **Gesamtarchitektur**
 - **Komponenten** und ausgewählte **Beispiele**
- **Diskussion** und Ausblick



Situationsabhängigkeit bei Microsoft-Produkten

Office Agent
„Büroklammer“

Was möchten Sie tun?
Ich verstehe Ihre Eingabe nicht.
Bitte formulieren Sie Ihre Frage neu.

Optionen Suchen

Office-Menü

Windows-Startmenü

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Situationsabhängigkeit bei Amazon.de

bestellen. Mehr dazu [hier](#).

KÜRZLICH ANGESEHEN

- [Hinter der Maske von Spider-Man - ein geheimer Blick hinter die hinter die Filmkulissen](#) von Mark Cotta Vaz
- [Elektronische Antriebstechnik. Dimensionierung von Antrieben mit Mathcad.](#) von Claus Wehrmann
- [Prozessorientierte betriebliche Informationssysteme...](#) von Michael Amberg 1)

Taschenbuch - 706 Seiten - Oldenbourg, Mchn.
Erscheinungsdatum: Oktober 2001
Auflage: 5. Aufl.
ISBN: 3486256351
Amazon.de-Verkaufsrang 917

Durchschnittliche Kundenbewertung: ★★★★★ 2)
 Anzahl der Kundenbewertungen: 2
[Schreiben Sie eine Online-Rezension](#) zu diesem Buch, und teilen Sie Ihre Gedanken anderen Lesern mit!

Kunden, die dieses Buch gekauft haben, haben auch diese Bücher gekauft: 3)

- [Bürgerliches Gesetzbuch \(BGB\)](#)
- [Praxishandbuch Industrietechnik und Informationstechnik \(Lernmaterialien\)](#) von Heinz O. Haberle
- [Mathematik für Ingenieure und Naturwissenschaftler, Bd 1](#) von Lothar Papula
- [Mathematik für Ingenieure und Naturwissenschaftler, Bd 2](#) von Lothar Papula

[Entdecken Sie](#)

Besonderheiten:

- **Kürzlich** verwendet/ individuelle **Favoriten**
- **Kommentare/** Rezensionen
- **Vergleich** mit anderen Anwendern

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Situationsabhängigkeit bei Google.de

Google-Suche: Gestaltung innovativer Nutzerschnittstellen - Microsoft Internet Explorer

Adresse: <http://www.google.de/search?hl=de&ie=UTF8&oe=UTF8>

Google

Erweiterte Suche Einstellungen Sprach-Tools Suchtipps

Gestaltung innovativer Nutzerschnittst Google-Suche

Suche: Das Web Seiten auf Deutsch Seiten aus Deutschland

Web Bilder Groups Verzeichnis

Das Web wurde nach Gestaltung innovativer Nutzerschnittstellen durchsucht. Resultate 1 - 10 von ungefähr 22. Suchdauer: 0.29 Sekunden.

[Wirtschaftsinformatik III, Prof. Michael Amberg, FAU Erlangen-Nürnberg](#)

... und aus den erkannten Situationen Informationen abzuleiten, die einen Mehrwert für Anwender (zB durch die Gestaltung innovativer Nutzerschnittstellen) und ...

www.wi3.uni-erlangen.de/forschung/fbci/ - 27k - [Im Archiv](#) - [Ähnliche Seiten](#)

Besonderheiten:

- Ergebnisse werden bereits **im Vorfeld** berechnet
- Steigerung der **Performanz** und damit der **Zufriedenheit** der Benutzer



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



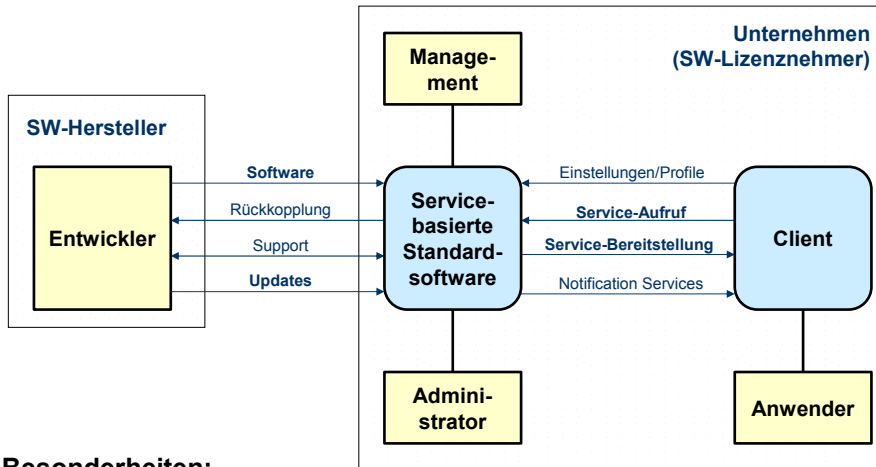
Agenda

- **Motivation**, Einführung und Einordnung
- **Situationsabhängige Adaption**
 - **Ziele** situationsabhängiger Adaption
 - **Szenarien** situationsabhängiger Adaption
- **Komponenten-Framework**
 - **Konzeption** und **Gesamtarchitektur**
 - **Komponenten** und ausgewählte **Beispiele**
- **Diskussion** und Ausblick



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Gesamtkontext adaptierbarer (Standard)Software



Besonderheiten:

- Mehrere **Interessensgruppen**
- Häufig **Multi-Tier-Architekturen**

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Adaptionskategorien und Interessensgruppen

Kategorien / Interessensgruppen	Adaptive Pull-Services	Adaptive Push-Services	Adaptive Nutzeroberfläche	Performance-management	Analyse, Wartung & Pflege	ASP & Intern. Leistungsverrechnung
End-anwender	X	X	X	X		
Administrator				X	X	
Management						X
Entwickler					X	X

Aber:

- **Datenschutzproblematik?**
- Wie viel ist **automatisierbar?**

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Szenarien adaptiver Software (1)

Adaptive **Pull**-Services:

- **Endgeräteabhängige Funktion**
- **Wertehistory** (Berücksichtigung der letzten Ansicht)

Adaptive **Push**-Services:

- **Workflow-Notification** (Email, SMS oder GUI)
- **Wert-Notification** (bei negativer Entwicklung eines Wertes)

Adaptive Benutzeroberfläche (**GUI**):

- **Personalisierte Benutzeroberfläche (GUI)**
- **Dynamische Menüs**



Szenarien adaptiver Software (2)

Nutzungsabhängige **Leistungsverteilung** und **Performancemanagement**:

- **Adaptives Caching**
- **Vorberechnung** (rechenintensiver Operationen)

Nutzungsabhängige Softwareanalyse, **-wartung**, und **-pflege**:

- **Prioritätsmanagement** (für die Weiterentwicklung)
- **Fehlermanagement** (Context eines Fehlers)

Interne Leistungsverrechnung und **ASP**:

- **Interne Leistungsverrechnung** (z.B. nach Nutzung)
- **Nutzungsanalyse** („Hineinschauen“ in eine Software)



Adaptionssystematik und Beispiele

Kategorien Granularität	Adaptive (Pull-) Services	Adaptive (Push-) Services	Adaptive Nutzer-oberfläche	Leistungsver- & Performance-management	Interne (Leistungs-) Verrechnung	Analyse, Wartung & Pflege	Erweitertes ASP
Service Request	Anpassung bezügl. Person, Zeit und Ort	Versenden einer Information (Notification Service), falls ein Wert einen definierten Bereich verlässt	Berücksichtigung von Userprofilen	Vorbereitung aller möglicher Anfragen	Separate Abrechnung beim Abrufen rechenintensiver Berichte	Fehler-meldung, Exception Management, Feedback bei Fehlern, Optimierung des Services im Zeitverlauf	Nutzungs-abhängige Abrechnung
Service	Werthehistory, Rezensionen, Empfehlungen, Notizen, Tips, Kommentare		Anpassung der Menüs, letzte Funktionen, Favoriten (clientseitig)	Individuelle kontextabhängige Vorbereitung (mittels Mustererkennung) oder Caching von lokalen Informationen	Interne Vergabe von Rechten für die Benutzung spezifischer Dienste		
Komponente			Anpassung der Default-GUI-Werte, Favoriten (serverseitig)	Lastverteilung, Verdichtungsoperationen	Abrechnung nach Benutzergruppen (die auf spezifische Komponenten zugreifen) und Verteilung der Kosten auf entsprechende Kostenstellen	Benutzungs-statistiken, Prioritätenmanagement für die Weiterentwicklung	
Applikation	Integration von Inhalten, Daten und Funktionen						

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

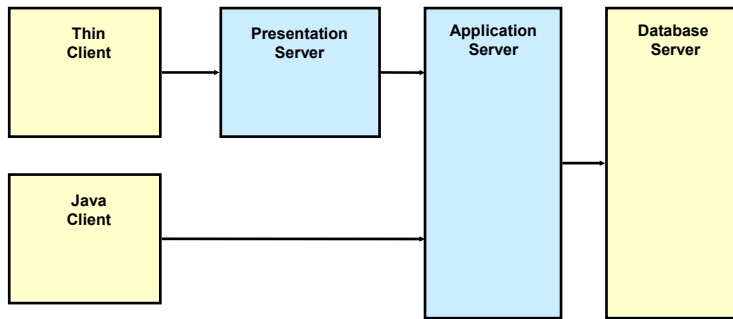
Agenda

- **Motivation**, Einführung und Einordnung
- **Situationsabhängige Adaption**
 - **Ziele** situationsabhängiger Adaption
 - **Szenarien** situationsabhängiger Adaption
- ➡ • **Komponenten-Framework**
 - **Konzeption** und **Gesamtarchitektur**
 - **Komponenten** und ausgewählte **Beispiele**
- **Diskussion** und Ausblick

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Anforderungen an Client/Server Basisarchitektur



Anforderungen:

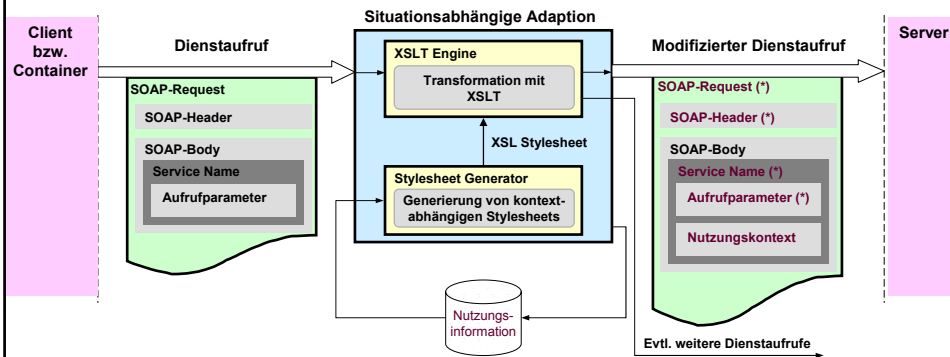
- **Lose Kopplung** zwischen Adaption und Kernsoftware
- **Flexibilität** hinsichtlich Adaption
- Gewährleistung **hoher Performanz**
- Realisierung mit **Standardkomponenten**
- Orientierung an **offenen Standards**



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Grundkonzept der situationsabhängigen Adaption



Anforderungen:

- **Lose Kopplung** zwischen Adaption und Kernsoftware
- Realisierung mit **Standardkomponenten**
- **Interceptor-Konzept** für vertikale (technische) Dienste ...



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Grundkonzept der situationsabhängigen Adaption

Anforderungen:

- Flexibles **Mapping** von Adaptionsoptionen
- Konzept des **Chain of Responsibility**
- Analog **Servlet Filtering** gemäß Spezifikation 2.4

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

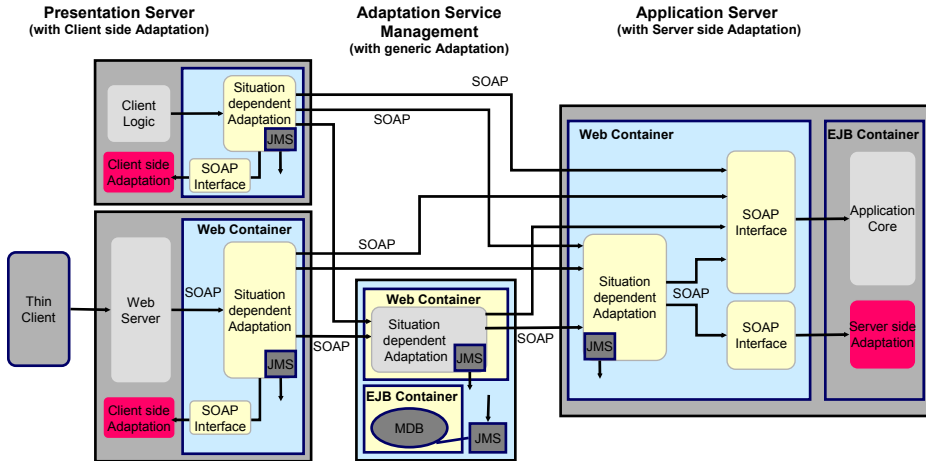
Grundkonzept der situationsabhängigen Adaption

Anforderungen:

- **Mehrstufiger Adaptionprozess**
- Flexibles **Pre-** und/ oder **Post-Processing**
- **Flexibilität** hinsichtlich Adaption

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Komponenten der Adaption

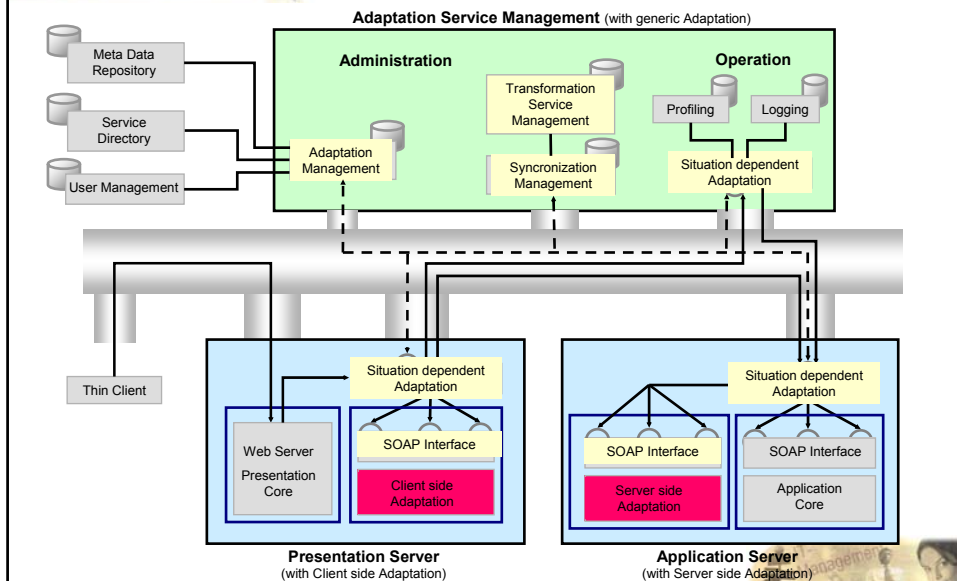


Anforderungen:

- **Mehrstufiger Adaptionprozess**
- **Performanz und Lastverteilung**

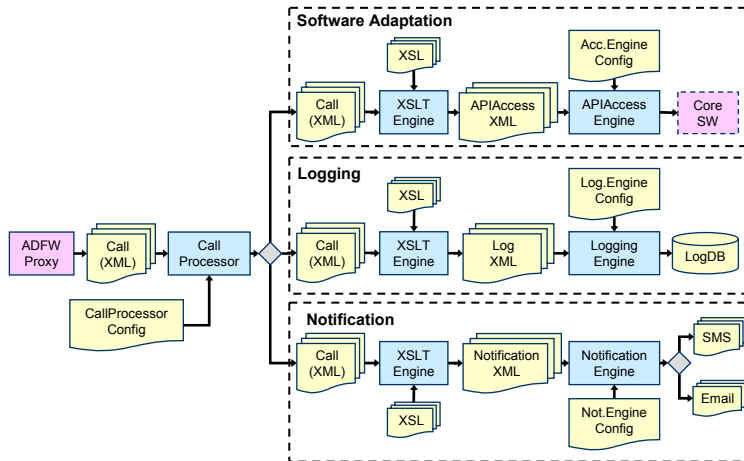
Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Gesamtarchitektur des Komponentenframeworks



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Bearbeitungsprozess

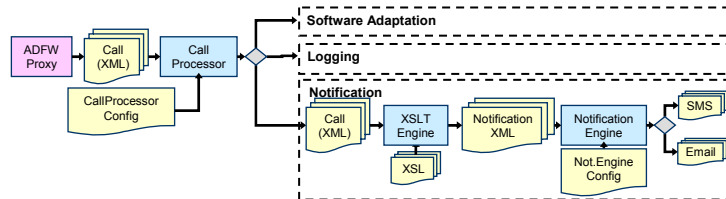


Anforderungen:

- **Flexibilität** in der Adaptierbarkeit
- **Abbildung Parallele** und **Sequenzielle Abläufe**

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Anwendung am Beispiel Controllingsoftware (1)

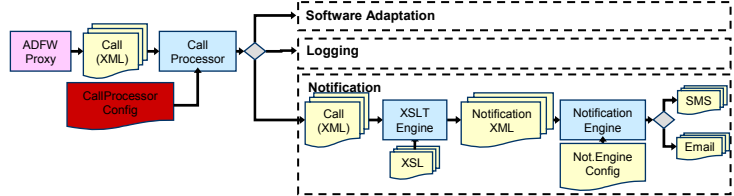


• Szenario:

- Ein **Administrator** ändert einige Benutzerdaten.
- Der **Benutzer** bekommt eine Benachrichtigung per email.
- Der **Benutzer** hat die Möglichkeit, die Änderungen auf Richtigkeit zu prüfen.

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Anwendung am Beispiel Controllingsoftware (1b)



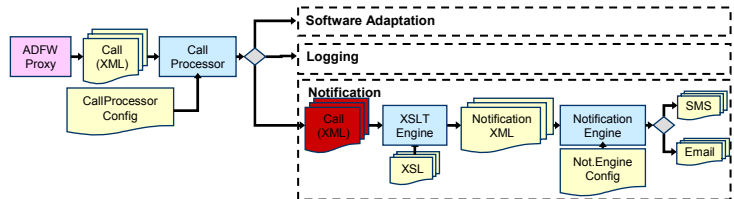
XML-Dokument zur Konfiguration der Bearbeitung:

```

<entry callId="PESLBUManagementBean.modifyUser" params="true">
  <parameters encode="partial">
    <parameter type="dto" class="com.prevero.everest.dto.PEDTOUser" name="user">
      <property type="simple" name="email"/>
      <property type="simple" name="phone"/>
      <property type="simple" name="mobile"/>
      <property type="simple" name="fax"/>
      <property type="simple" name="firstName"/>
      <property type="simple" name="name"/>
    </parameter>
  </parameters>
  <pipe>
    <pi engineId="notification" xsl="xmlUserModifiedNotification.xslt"/>
  </pipe>
</entry>
<engine engineId="notification" engineClass="com.prevero.everest.adaption.notification.NotificationEngine"/>

```

Anwendung am Beispiel Controllingsoftware (1c)



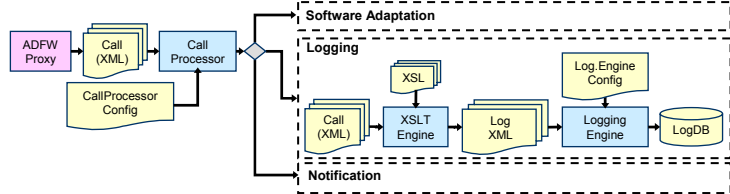
XML-Dokument des Serviceaufrufs:

```

<call timeStamp="1043944771703" serviceName="PESLBUManagementBean.modifyUser">
  <ejbName>PESLBUManagementBean</ejbName>
  <methodName>modifyUser</methodName>
  <userId>1</userId>
  <parameters>
    <user>
      <email>mail@okujava.de</email>
      <phone>+499113603188</phone>
      <mobile>+491728318203</mobile>
      <fax>+499113603188</fax>
      <firstName>Shota</firstName>
      <name>Okujava</name>
    </user>
  </parameters>
</call>

```

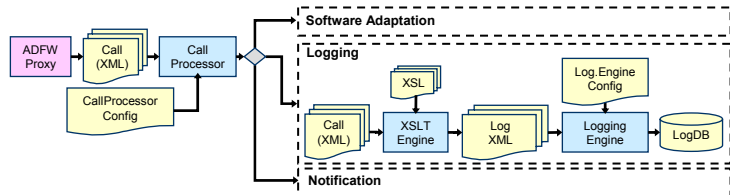

Anwendung am Beispiel Controllingsoftware (2)



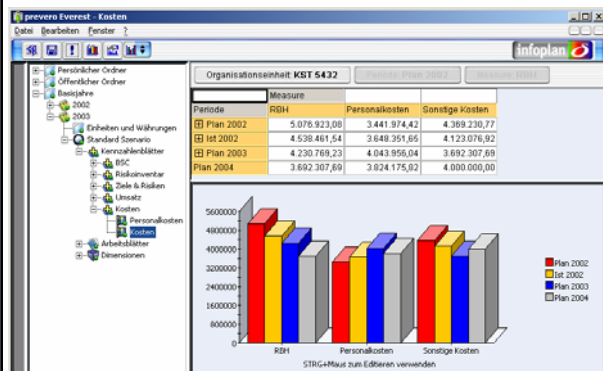
- **Szenario:**
 - Ein **Benutzer** möchte seine/ die **Favoriten** von Bearbeitungsfenstern bzw. Funktionen sehen.

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Anwendung am Beispiel Controllingsoftware (2b)



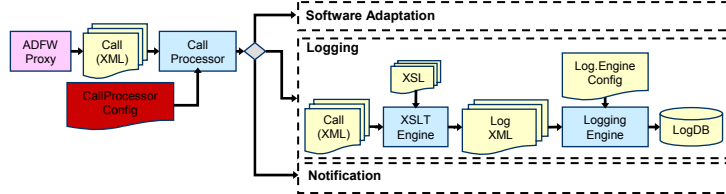
Kernsoftware:



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Anwendung am Beispiel Controllingsoftware (2c)

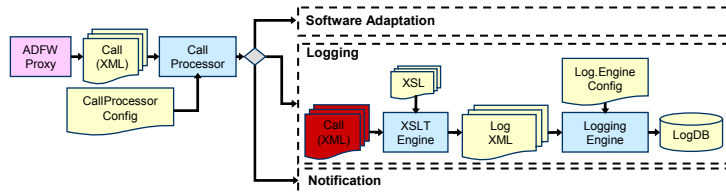


XML-Dokument zur Konfiguration der Bearbeitung:

```
<engine engineId="logger" engineClass="com.prevero.everest.adaption.Logging.LoggingEngine"/>
<entry callId="com.prevero.everest.server.view.ejb.PESLBViewManagementBean.getView()" params="true">
  <parameters encode="partial">
    <parameter type="dto" class="com.prevero.everest.dto.PEDTOView" name="valueObject">
      <parameter type="simple" name="ID"/>
      <parameter type="simple" name="scenarioID"/>
      <parameter type="simple" name="measureSheetID"/>
      <parameter type="simple" name="timeBaseID"/>
      <parameter type="simple" name="name"/>
    </parameter>
  </parameters>
  <pipe type="request">
    <pi engineId="logger" xsl="xml/CountVisit.xslt"/>
  </pipe>
  <pipe type="request">
    <pi engineId="logger" xsl="xml/getTop10.xslt"/>
  </pipe>
</entry>
```



Anwendung am Beispiel Controllingsoftware (2d)



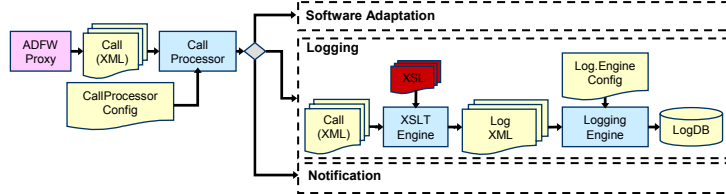
XML-Dokument des Serviceaufrufs:

```
<call timeStamp="1046180692641"
serviceName="com.prevero.everest.server.view.ejb.PESLBViewManagementBean.getView()"
  <ejbName="com.prevero.everest.server.view.ejb.PESLBViewManagementBean"</ejbName>
  <methodName="getView()"</methodName>
  <userId=0</userId>
  <parameters>
    <valueObject>
      <ID>1162</ID>
      <scenarioID>1</scenarioID>
      <measureSheetID>1</measureSheetID>
      <timeBaseID>1</timeBaseID>
      <name>Ziele</name>
    </valueObject>
  </parameters>
</call>
```





Anwendung am Beispiel Controllingsoftware (2e)



XSLT-Dokument zu *CountVisits*:

```

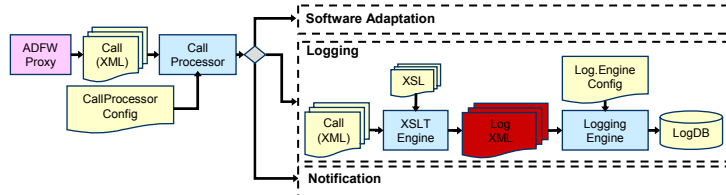
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="call">
    <logroot>
      <entry target="countVisits" action="UPDATE">
        <primaryKey id="viewid" value="{parameters/valueObject/ID}" type="number"/>
        <primaryKey id="userid" value="{userid}" type="number"/>
        <data id="td" type="date"><xsl:value-of select="@timeStamp"/></data>
        <data id="name" type="string"><xsl:value-of select="name"/> </data>
        <data id="userid" type="number"><xsl:value-of select="userid"/></data>
        <data id="viewid" type="number"><xsl:value-of select="ID"/></data>
        <data type="counter" id="ct"/>
      </entry>
    </logroot>
  </xsl:template>
</xsl:stylesheet>

```

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg



Anwendung am Beispiel Controllingsoftware (2f)



XML-Dokument für die Datenbankoperation:

```

<logroot>
  <entry action="UPDATE" target="countVisits">
    <primaryKey type="number" value="1162" id="viewid"/>
    <primaryKey type="number" value="0" id="userid"/>
    <data type="date" id="td">1046180692641 </data>
    <data type="string" id="name">Ziele </data>
    <data type="number" id="userid">0 </data>
    <data type="number" id="viewid">1162 </data>
    <data id="ct" type="counter"/>
  </entry>
</logroot>

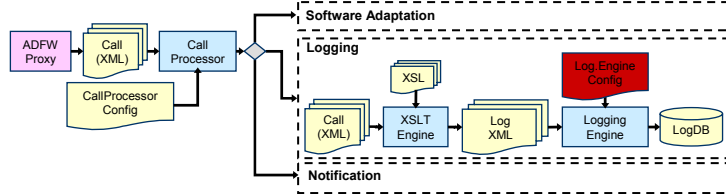
```

Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg





Anwendung am Beispiel Controllingsoftware (2g)



XML-Dokument zur Konfiguration der Datenbankoperation:

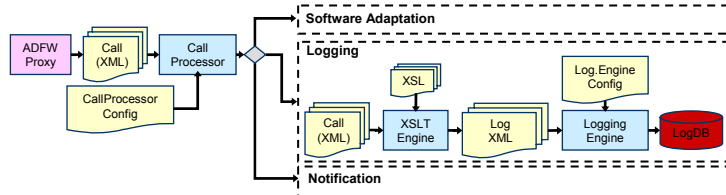
```

<preferences EXTERNAL_XML_VERSION="0.0">
  <root type="user">
    <node name="adfw">
      <node name="logging">
        <map>
          <entry key="driver" value="org.gjt.mm.mysql.Driver"/>
          <entry key="dbURL" value="jdbc:mysql://WKS-OKUJAVA:3306/everest_log"/>
          <entry key="userName" value="dummy"/>
          <entry key="pwd" value="dummy"/>
        </map>
      </node>
    </map>
  </root>
</preferences>

```



Anwendung am Beispiel Controllingsoftware (2h)



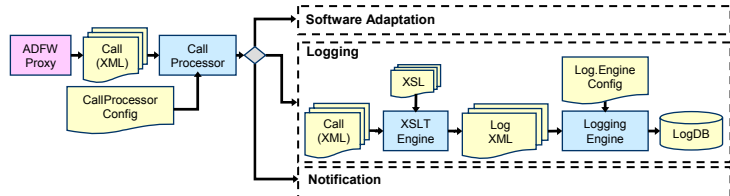
Datenbankeintrag:

```
mysql> select * from countvisits where userId=0;
```

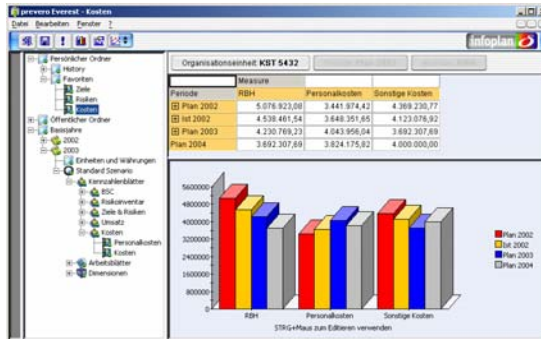
viewId	userId	ct	td	name
1162	0	12	2003-02-25	Ziele
121	0	10	2003-02-25	Risiken
113	0	9	2003-02-25	Kosten
145	0	1	2003-02-25	Tagesabsatz



Anwendung am Beispiel Controllingsoftware (2i)



Kernsoftware (mit Favoriten):



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg

Implementierung von Adaptionsszenarien

ID	Name	Klasse		Interessengr.	User - io				Darstellung			Log.	# User	Z	Trigger	Au												
		adapt. Push	adapt. GUI	Performancem.	ASP & int. LV	User	Management	Admin	Entwickler	Designer	Manager	Poweruser	Kern-SW (ohne GUI)	Kern-SW (nur GUI)	Email	SMS	Admin-Tool	mit	ohne	Einzeluser	Gruppe	Alle	Request	Response	Zeit	Aufwand		
120	konkrete Zelle überschreitet Wert / verlässt Zielbereich	X				X			io	io	io	io	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
121	Darstellung der internen LV (absolute Kosten)				X	X	X		io	io	io	io	X					X	X	X	X	X	X	X	X	X	X	0
122	Darstellung der internen LV																	X	X	X	X	X	X	X	X	X	X	-
120a	beliebige Zelle überschreitet einen Wert																	X		X	X	X	X	X	X	X	X	+
120b	beliebige Zelle in einem konkreten Arbeitsblatt überschreitet einen Wert																	X		X	X	X	X	X	X	X	X	+
120c	konkrete Zelle überschreitet einen Wert																	X		X	X	X	X	X	X	X	X	+
120d	konkrete Zelle überschreitet einen Wert																	X		X	X	X	X	X	X	X	X	+
120e	konkrete Zelle verlässt einen definierten Zielbereich																	X		X	X	X	X	X	X	X	X	+
120f	konkrete Zelle verlässt einen definierten Zielbereich																	X		X	X	X	X	X	X	X	X	+
121a	Darstellung der internen LV	öffentlich																X		X	X	X	X	X	X	X	X	+

Übersicht verschiedener Szenarien

Detailsicht des „Szenario 120“



Agenda

- **Motivation**, Einführung und Einordnung
- **Situationsabhängige Adaption**
 - **Ziele** situationsabhängiger Adaption
 - **Szenarien** situationsabhängiger Adaption
- **Komponenten-Framework**
 - **Konzeption** und **Gesamtarchitektur**
 - **Komponenten** und ausgewählte **Beispiele**
- • **Diskussion** und Ausblick



Diskussion und Ausblick

- **Wie sehen zukünftige Anwendungen** und der **Mehrwert** adaptiver Software aus?
- **Wie kann man Potenziale** der Adaption erkennen?
- **Wie können „neue“** und „intelligente“ Formen der **Rückkopplung** gestaltet werden?
- **Kann man in Software „hineinschauen“?**
- **Ist die Adaptivität von Software ein Grundbestandteil von evolutionären (Software-)Systemen?**





Ausblick



Komponenten-Framework für situationsabhängige Adaption Web-Service-basierter Standard-SW, FAU Erlangen-Nürnberg