

Grobe Auflistung der Aufgaben für den Feinentwurf und Nachfolgendes

18. November 2009

1 Gui Team

1.1 Verantwortliche

- Feras
- Daniel

1.2 Evaluieren

- Klären wie das DM-Plugin und speziell die DM-Property-GUI aufgebaut sein soll, damit sie auch in allen Belangen erweiterbar ist.
- Untersuchen ob dafür Extension Points möglich sind bzw. es welche dafür gibt: (die Plug-Ins sollen möglichst einfach erweitert werden können)
 - Extension Points für weitere Punkte in der Admin-Konsole (zB Referenzen)
 - Extension Points für weitere DM-Aktivitäten
 - Extension Points für weitere Patterns
 - Extension Points für weitere Anfragesprachen und ihre Komponenten (zB. SQL: SELECT, FROM, WHERE, ORDER BY usw.)
- Allgemein das Thema Erweiterbarkeit untersuchen, die ist am wichtigsten. Am Besten überlegen was sich alles ändern würde, wenn da jetzt eine ganz neue Aktivität realisiert werden soll, mit einer ganz anderen Anfragesprache usw..
- Die extra BPEL Variablen die benötigt werden (welche? für was?) und wie diese modellierungstechnisch (im BPEL Designer) umgesetzt werden können (Mirko meinte man müsse dazu den BPEL Designer evtl. aufbohren, das wollen möglichst verhindern)
- Überprüfen ob "External Variables" (siehe Apache ODE) im BPEL Designer unterstützt werden bzw. unterstützt werden können (nur für den Fall, dass wir diese benutzen. Muss das ODE Team erst klären)
- Transformation vor Deployment möglich ohne den BPEL Designer aufzubohren?
- Verbindung zum SIMPL Core, wie? wo? als extra Plug-In-Komponente?

1.3 Machen

- Konzept und Diagramm für Aufbau der GUI und DM-Properties mit ersichtlichen Extension Points
- Klassendiagramm für den Aufbau der DM-Properties (siehe Michael Hahn) und der GUI
- Dynamic UML (Sequenz-, Ablaufdiagramm, ...) von Modellierung mit einer DM-Aktivität, ggf. evaluieren der Eingaben, erzeugen des erweiterten BPEL-Codes, Auflösen der Referenzen und Deployment

2 Apache Ode Team

2.1 Verantwortliche

- René
- Wolfgang

2.2 Evaluieren

- Ist es möglich die Auditing-DB mit einem DAO auf den SIMPL Core umzubiegen um dort eine RDB-Datenquelle zu verwenden?
- Events direkt in EA abfeuern möglich? Werden diese beim Auditing erfasst?
- Wie und wo wird die Verbindung zum SIMPL Core hergestellt (je nachdem wo er ist. Muss das SIMPL Core Team erst klären)
- Plug-In-System für zusätzliche DM-Aktivitäten (glaube das ist das einfach nur eine JAR hinzufügen, evtl. können wir da aber auch was eigenes abstrahieren)
- Einsatz von External Variables überprüfen (siehe ODE Homepage)
- DOM Parser für EA Elemente (evtl. gibts da schon was in ODE oder was extra, evtl. Open Source)
- Wie soll die DOM-Element-spezifische Ausführung realisiert werden?
- Kann der Deployment Descriptor um eigene Granularität erweitert werden? Und ist das irgendwie zur Laufzeit änderbar?

2.3 Machen

- Test einer EA mit Event abfeuern (siehe Christoph Müller)
- Klassendiagramm zu EA, Verbindung zu SIMPL Core, Eventsystem und DAO
- Dynamic UML (Sequenz-, Ablaufdiagramm, ...) von deployten Prozess starten, EA-Activity einlesen, EA-DOM parsen, Ausführung über den SIMPL Core, Ergebnis in Prozess

3 Simpl Core Team

3.1 Verantwortliche

- Michael
- Schneimi
- Xi

3.2 Evaluieren

- Extra Axis2 oder innerhalb ODE Axis2 (Entscheidung wichtig!)
- Schnittstellen definieren, Welche Services müssen welche Funktionen bereitstellen
- AdministrationService wird evtl. nicht gebraucht, andere Services können direkt benutzt werden
- Wie soll der erweiterte BPEL-Code aussehen, muss noch spezifiziert werden. (evtl. zusammen mit GUI Team)
- Gibt es Zustände im SIMPL Core, die evtl. auch wegen Transaktionen gesichert werden müssen? (persistenz über die interne DB?)
- StorageService mit interner Derby DB? Evtl. besser als über den DatasourceService.

3.3 Machen

- Grobentwurf überarbeiten
- evtl. SDO, DAS Testbeispiel (siehe Michael Hahn)
- Klassendiagramm für jeden Service
- Dynamic UML (Sequenz-, Ablaufdiagramm, ...) für Abläufe bei denen sich die Services gegenseitig benutzen und zB auch Ablauf innerhalb des DatasourceService, bzw. dort wo es dann halt komplexer wird.