Manuel Schönberner *06.04.2012*

Entwurfsbeschreibung des Gesamtprojekts

Swp 12-2 ColaNut

1.Allgemeines

Die zu entwickelnde Web-GUI stellt eine Bedienungsoberfläche für das vorhandene und auf Java basierende Programm LIMES dar. Sie läuft Serverseitig, sodass nur minimale Datenmengen zum Client gesendet werden müssen. Als Clients kommen, im Allgemeinen, alle gängigen Browser in Frage. Die Web-GUI sieht eine Nutzerverwaltung und eine Datenhaltung von individuellen Nutzerdaten vor, welche in zukünftigen Releases ausgebaut und funktionstüchtig eingefügt wird. Weiterhin werden alle derzeitigen Funktionen von LIMES unterstützt und können problemlos aufgerufen und ausgeführt werden. Eine Live-Demo des Vorprojekts ist unter colanut.no-ip.org:9000 *zu finden.*

2.Produktübersicht

Die Web-GUI wird für eine vereinfachte Nutzung von LIMES erstellt und soll dieses Programm zugänglicher und intuitiver gestalten. So werden verschiedene Tabs zur Gruppierung von LIMES internen Funktionen genutzt. So wird ein Tab für den „Link-Spec“ an sich erzeugt, in welchem alle weiteren eingebettet sind. Ein weiterer für die Metric, welcher alle GUI-Elemente zur Erstellung besitzt, sowie ein Tab welcher für die Erzeugung der XML dient. Ein weiterer Tab beinhaltet die Matches. Ebenfalls wird ein Learning-Tab vorhanden sein, welcher alle Möglichkeit für active- bzw. Batch–learning beinhaltet zuletzt ein Tab für das Review. Weiterhin können verschiedene Methoden zur Bearbeitung der „LinkSpec“ - Eigenschaften per Drag and Drop zusammengefügt werden. Ein Log-In Screen wurde zum Zugriff auf die Software ebenfalls eingefügt, welcher die Grundlage für die Erweiterung zur Nutzerverwaltung bildet. Das vorhandene Design wird voraussichtlich in zukünftigen Releases im Grundlegenden beibehalten, mit Ausnahme kleinerer, während des Arbeitsverlaufs auffallender Verbesserungen der Ansicht, sowie Verbesserungen der Nutzbarkeit. Die bereits erwähnte Userverwaltung wird mithilfe einer Datenbank realisiert. Weitere Funktionen der GUI beinhalten das Hochladen von LinkSpecs, sowie das Runterladen der Ergebnisse. Da die Web-GUI

3.Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem / der einzelnen Pakete[1]

Die Web-GUI wird mit dem Play! Webframework erstellt. Dieses Webframework basiert auf Java und HTML, besitzt allerdings eine Framework eigene „vereinfachte“ Syntax. Dieses Framework basiert auf dem Model-View-Controller Designpattern. Es gibt eine große Auswahl an Plug-Ins und Erweiterung für dieses Framework sowie eine einfachere Nutzung verschiedener Datenbanken. Die angestrebte Datenbank für die Web-GUI wird Apache-Derby werden. Im ersten Release wird diese Datenbank aufgesetzt, aber der Zeitpunkt an dem die Nutzerverwaltung und das sichern der Dokumente, für welche die Datenbank vorgesehen ist, voll funktionstüchtig ist, kann noch nicht abgesehen werden. Die Zugriffe auf diese Datenbank werden über Java geschehen und über Buttons an die GUI gebunden. So zum Beispiel das Anlegen neuer User, eine Passwortabfrage oder aber das speichern von Dokumenten (z.B. Auflistung der Matches oder LinkSpecs). Das Ausführen der LinkSpecs geschieht auch per Button-klick. Ein wichtiger Aspekt ist die dem Play!-Framework eigene Syntax, welche eine vereinfachte Programmierung der GUI zulässt. Play! besitzt ebenfalls eine eigene, festvorgegebene Ordnerstruktur, deren Darstellung im Anhang zu finden ist. Vertiefend ist weiterhin zu erwähnen, dass zur allgemeinen Gestaltung der GUI ebenfalls jQuery UI 1.8.13 genutzt wurde. Dies ist ein Toolkit, welches auf der JavaScript jQuery – Bibliothek aufgesetzt wurde und für low-level Interaktion und Animation genutzt werden kann. Dies wurde auch für die Erstellung der Tabs, welche zur Gruppierung genutzt werden und für die eingebetteten Buttons verwendet. Für die Darstellung und Funktion der Metric fand jsPlumb, ebenfalls auf der jQuery – Bibliothek basierend, Einsatz, welches eine Drag and Drop – Funktion mit sich bringt. Zur Kommunikation der GUI mit Java Applikationen (in diesem Fall LIMES) wurde JavaScript Object Notation (JSON), genauer JSON2 genutzt und dazu GSON, da zum einen GSON eine Umwandlung von JSON – Objekten in Strings und umgekehrt ermöglicht und JSON ein einfach zu lesendes, schreibendes und parsendes Datentauschformat. Die Grundlegende Basis bildet, im Allgemeinen, die bisherige Version von ColaNut, welche komplett auf JavaScript basiert. Ansonsten sind noch die LIMES-Bibliotheken enthalten, welche für die Funktionalität von LIMES unerlässlich sind. Das Hoch- und Runterladen von Dokumenten (Ergebnisse bzw. LinkSpecs) wird vorrausichtlich per http-script geschehen und in der GUI werden Buttons zu finden sein, über welche

---

1 Alle Adressen zur Internetpräsenz der genannten Programme, Plug-Ins oder ähnliches findet sich im Anhang wieder

diese angesteuert werden können. Es wird also nach dem Klick ein Dialog gestartet (sowohl beim Down- als auch Upload) i´n welchem man die Zieldatei / Zielpfad wählen kann. Jedoch sollte beachtet werden, dass sich im Laufe der Implementierungszeit die Mittel zur Erstellung der Web-GUI ändern können, da auf Kompatibilität des gesamt Projekts hingearbeitet wird.

Anhang

www.**playframework**.org .. Play! Framework
db.apache.org › db .. Apache Derby
jsplumb.org/ .. JSPlumb
jqueryui.com .. jQuery UI

Darstellung Ordnerstruktur Play!

```
app                     → Application sources
 └ assets               → Compiled asset sources
    └ stylesheets       → Typically LESS CSS sources
    └ javascripts       → Typically CoffeeScript sources
 └ controllers          → Application controllers
 └ models               → Application business layer
 └ views                → Templates
conf                    → Configurations files
 └ application.conf     → Main configuration file
 └ routes               → Routes definition
public                  → Public assets
 └ stylesheets          → CSS files
 └ javascripts          → Javascript files
 └ images               → Image files
project                 → sbt configuration files
 └ build.properties     → Marker for sbt project
 └ Build.scala          → Application build script
 └ plugins.sbt          → sbt plugins
lib                     → Unmanaged libraries dependencies
logs                    → Standard logs folder
 └ application.log      → Default log file
target                  → Generated stuff
 └ scala-2.9.1
    └ cache
    └ classes           → Compiled class files
    └ classes_managed   → Managed class files (templates, ...)
    └ resource_managed  → Managed resources (less, ...)
    └ src_managed       → Generated sources (templates, ...)
test                    → source folder for unit or functional tests
```