



LinkedGeoData.org

— Projektangebot —

Von: Peter Pfahl (Projektleiter)

Abgabe: 27. Januar 2012

Gruppe: SWP12-11



Das von uns zu erstellende Softwareprodukt soll eine Benutzerschnittstelle zur Verlinkung von Daten aus zwei Datenbanken, konkret aus LinkedGeoData¹ (LGD) und DBpedia², bilden. Da die eigentliche Verknüpfung durch einen bereits gegebenen Algorithmus geschieht, konzentriert sich dieses Projekt hauptsächlich auf die Benutzeroberfläche (GUI) und die benutzerseitige Kontrolle der Verknüpfungsergebnisse. Hierfür bestehen in erster Linie besondere funktionale Anforderungen an die GUI.

Funktionale Anforderungen

Zu Beginn erfolgt einer Anmeldung durch ein registriertes Benutzerkonto, welche durch die Plattform organisiert wird, in der das Produkt ausgeführt wird.

Der Nutzer soll anschließend auf einer Karte, in der er frei zoomen kann, einen rechteckigen Bereich auswählen können, in dem er nun mit dem Produkt arbeitet. Diese Auswahl bleibt auch bei veränderter Zoomstufe stabil.

Zunächst kann der Benutzer die anzuzeigenden Daten aus den 2 Datenbanken wählen, sodass die Ontologien auf der Karte farblich differenziert zu sehen sind. Ebenso kommt mit der Wahl der Ontologien auch der Algorithmus ins Spiel, der nun automatisch alle im Kartenausschnitt angezeigten Objekte auf mögliche Verknüpfungen überprüft und diese gegebenenfalls vornimmt. Das Ergebnis ist nun ein Kartenabschnitt, der die ausgewählten Daten der beiden Datenbanken darstellt und auch optisch verknüpft. Diese Verknüpfungen erscheinen zusätzlich in einer Liste am unteren Bildschirmrand und werden sowohl auf der Karte als auch in der Liste hervorgehoben, wenn der Mauszeiger auf sie zeigt. Somit ist eine eindeutige Zuordnung der Verknüpfungen auf der Karte zu denen in der Liste gewährleistet.

Nun beginnt die Hauptaufgabe des Benutzers. Er muss nun überprüfen, ob die vom Algorithmus ausgegebenen Verknüpfungen korrekt sind oder nicht. Hierfür arbeitet er die angezeigten Verknüpfungen nacheinander durch und entscheidet bei jeder einzelnen, ob sie seiner Meinung nach richtig oder falsch ist. Ebenso hat er die Möglichkeit, neue Verknüpfungen hinzu zu fügen und fehlerhafte nach Ablehnung zu löschen. Der Status einer Verlinkung (richtig, falsch, ungeprüft) soll dabei in der Liste farblich erkennbar sein.

Weitere Anforderungen ergeben sich in den Bereichen der Datenverwaltung und des Linkings.

So ist es wichtig, die Anzahl der angezeigten Objekte zu limitieren, um in sinnvollem Maßstab zu arbeiten und die Performance zu sichern. Welche Methode der Limitierung für unser Produkt am effektivsten ist wird noch überprüft werden. Es gibt die Möglichkeit der Limitierung der Zoomstufe, so dass nur eine gewisse Anzahl an Ontologien gezeigt werden. Eine andere Möglichkeit wäre auf beliebiger Zoomstufe nur eine maximale Anzahl an Datensätzen anzuzeigen.

Ein weiterer Performancegewinn ist zu erhoffen, wenn beim Linking die Daten schon vorgefiltert und eventuell auch sortiert übergeben werden.

¹<http://linkedgeodata.org>

²<http://dbpedia.org>



Es sollen alle Änderungen der Benutzer in einer History gespeichert werden. Da der Historyeintrag auch den Nutzernamen enthält sind diese zu einem späteren Zeitpunkt eindeutig zu einem Benutzer zuordenbar. Dies ist wichtig, um unerwünschte Manipulationen der Daten auf ihren Urheber zurück zu führen und damit eventuelle negative User und Missbrauch aufzudecken.

Desweiteren haben wir Anforderungen definiert, die **optional** bei ausreichender Zeit oder von nachfolgenden Entwicklern erfüllt werden können.

So besteht unter anderem die Idee, eine kreisförmigen Auswahl des zu bearbeitenden Bereiches zusätzlich zur rechteckigen zu implementieren. Auch eine Sortierung der Verlinkungen in der Liste nach verschiedenen Kriterien und eine alternative GUI mit work-in-map Konzept sind als Optionen geplant. Zur Erweiterung der Benutzerkonten-Funktion wären außerdem eine Statistik nach Benutzer und eine Kommentarfunktion zu Änderungen wünschenswert.

Nichtfunktionale Anforderungen

Auf der Seite der nichtfunktionalen Anforderungen liegt das Gewicht deutlich auf Kompatibilität und Ergonomie, die beide mit „wichtig“ eingestuft sind. Die Kompatibilität ist bei dieser Internet-Anwendung vor allem wegen der Verwendung verschiedener Browser von großer Bedeutung. Außerdem muss sich die Anwendung möglichst einfach in eine bereits bestehende Internetseite einbinden lassen.

Der hohe Stellenwert der Ergonomie ergibt sich aus dem Anwendungsszenario der Software. Da der Nutzer **freiwillig** seine Zeit damit verbringt, die Daten zu verknüpfen, zu validieren und zu kontrollieren, müssen ihm diese Arbeitsschritte so einfach und intuitiv wie möglich gemacht werden.

Mittleren Qualitätsanspruch stellen wir an die Erweiterbarkeit und Performance des Programmes. Die zu bearbeitenden Aufgaben sind nicht besonders rechenintensiv und eine konkrete komplexere Erweiterung noch nicht in Aussicht. Da es sich nicht um eine sicherheitsrelevante Anwendung handelt, stehen auch die Anforderungen an Sicherheit und Stabilität auf einem normalen Niveau.



Projektablauf

Um das Projekt wie geplant umzusetzen, bedarf es einer guten Planung des Projektablaufes. Hierfür werden die einzelnen Arbeitsschritte in grobe Arbeitspakete unterteilt.

Zu Beginn steht noch die **Konkretisierung der Anforderungen** im Fokus. In Gesprächen mit den Betreuern geschieht eine Einigung und klare Festlegung der Anforderungen, bei der Missverständnisse vermieden werden müssen. Erste Modellierungen über den Softwareablauf unterstützen diese Phase. In einem **ersten Meilenstein** werden das hier beschriebene Projektangebot sowie das Vorprojekt vorgestellt, diskutiert und ausgewertet, um es im Nachhinein zu verbessern.

Der Aufwand dieses Arbeitspaketes wird mit 15% vom Gesamtprojekt beziffert.

Diesem Optimierungsprozess soll anschließend der **Projektvertrag** als Resultat folgen, unterstützt von einer ersten **Softwarestudie**. Sie soll bereits eine funktionale Muss-Anforderung umsetzen und demonstrieren. Beide Ergebnisse dieser Phase sollen in einem **zweiten Meilenstein** den Betreuern präsentiert werden.

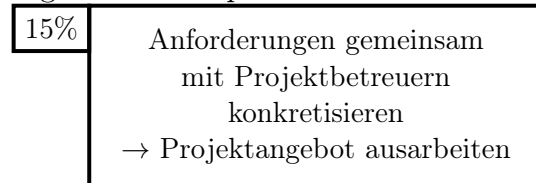
Der Aufwand dieses Arbeitspaketes wird mit 15% vom Gesamtprojekt beziffert.

Nun folgt die Kernphase des Projektes, die **Planung und Durchführung der Implementierung**. Hierfür wird die zu erstellende Software in sinnvolle Teilprojekte gegliedert, die im Wochenzyklus stabil abgeschlossen werden sollen. Es werden vorrangig die Muss-Anforderungen implementiert. Programmteile werden immer lokal vom Programmierer durch UnitTests getestet, sodass sich nur lauffähiger Code in den Repositories befindet. Der Fortschritt ist durch die wöchentlichen Teilprojekte und 2 geplante Release Candidates (RC1 und RC2) in dieser Phase gut zu beobachten.

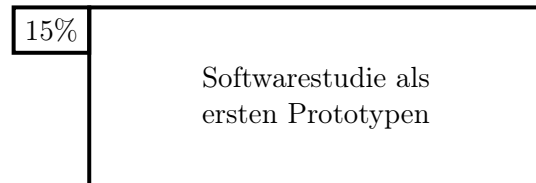
Der Aufwand dieses Arbeitspaketes wird mit 40% vom Gesamtprojekt beziffert.

In der nächsten Phase überschneiden sich zwei Aufgabenbereiche.

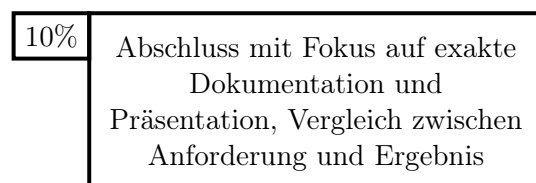
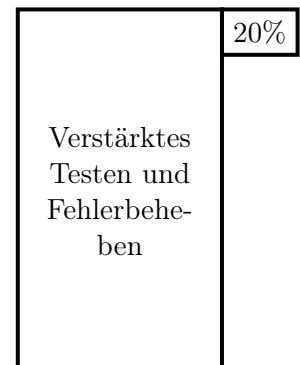
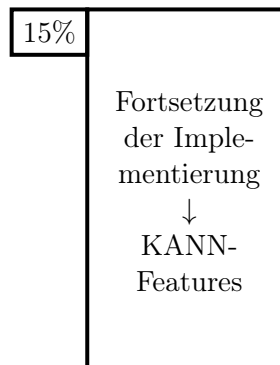
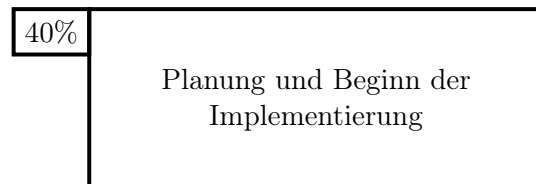
Auf der einen Seite kümmert sich die **Fortsetzung der Implementierung** um die noch fehlenden Muss-Features und kann bei ausreichend Zeit bereits mit der Umsetzung von Kann-Anforderungen beginnen. Ihre Ergebnisse werden durch einen dritten Release Can-



1. Meilenstein



2. Meilenstein





didate (RC3) repräsentiert.

Der Aufwand dieses Arbeitspaketes wird mit 15% vom Gesamtprojekt beziffert.

Parallel dazu rückt das **verstärkte Testen und Bugs fixen** in den Vordergrund. Stabilität und Ergonomie haben hier Priorität vor Perfektionismus. Das Ergebnis soll ein vierter Release Candidate (RC4) sein, der dem Endprodukt schon weitestgehend entspricht.

Der Aufwand dieses Arbeitspaketes wird mit 20% vom Gesamtprojekt beziffert.

Das Ende macht der **Abschluss** mit besonderem Augenmerk auf die externe Dokumentation und die Präsentation der Software. Außerdem sollen die erstellten Anforderungen aus dem Pflichtenheft mit dem tatsächlichen Ergebnis verglichen werden.

Der Aufwand dieses Arbeitspaketes wird mit 10% vom Gesamtprojekt beziffert.

Der **Gesamtaufwand** des Softwareprojektes ergibt damit 115% und beinhaltet auch die Implementierung der Kann-Anforderungen. Diese könnten jedoch auch nachträglich von zukünftigen Projektgruppen oder anderen Dritten umgesetzt werden.



Programmablaufskizze

