

BioKemika

ELearning-Projekt des
Fachbereichs Biochemie, Chemie und
Pharmazie (FB13)

Sven Köppel
Technischer Leiter

Goethe-Universität Frankfurt



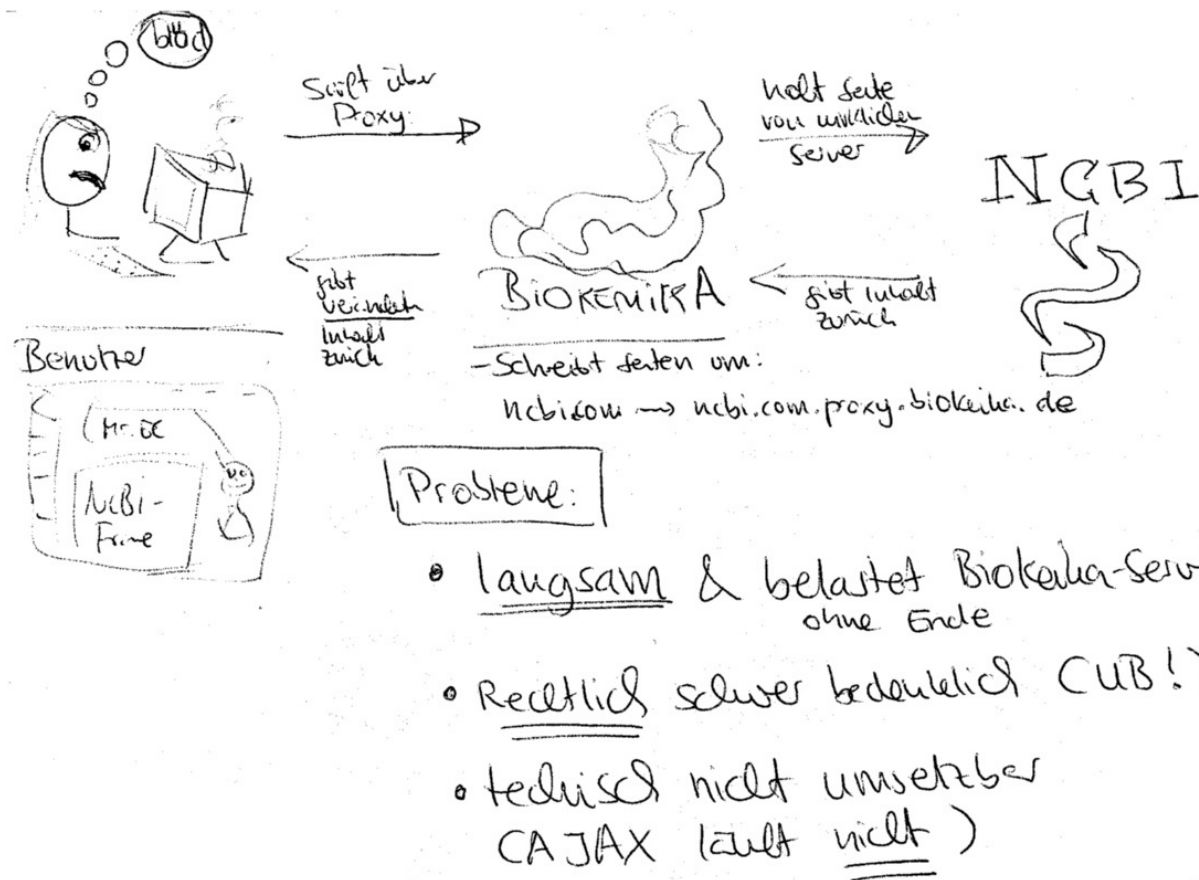
Sven Köppel

MetaSearch 2010

BioKemika goes social

Was ist schlecht an MetaSearch?

Zunächst eine Skizze, wie MetaSearch 2009 funktionierte und wo die Probleme waren:



und nun ein zusammenfassendes Zitat aus der E-Mail „Epochentag“ von mir:

Der MetaSearch-Proxy ist eine Sackgasse. Nicht (nur) wegen Geschwindigkeit oder den rechtlichen Problemen, sondern vor allem aus technischer Seite: „AJAX/Web 2.0-Seiten gehen damit einfach nicht. Und das sind sehr viele.“

Mehr bleibt nicht zu sagen.

Grunkonzept von MetaSearch 2010

Schon die alte MetaSearch entsprach in der Funktion nicht mehr ihrem Namen. Die neue tut dies noch viel weniger. Daher ist „MetaSearch 2010“ nur als Entwicklungsname zu verstehen, so wie „Longhorn“ sogar im Sprachgebrauch für „Windows Vista“ gebraucht wurde.

Auch hier eine kleine Skizze, wie sich die neue MetaSearch verhält:

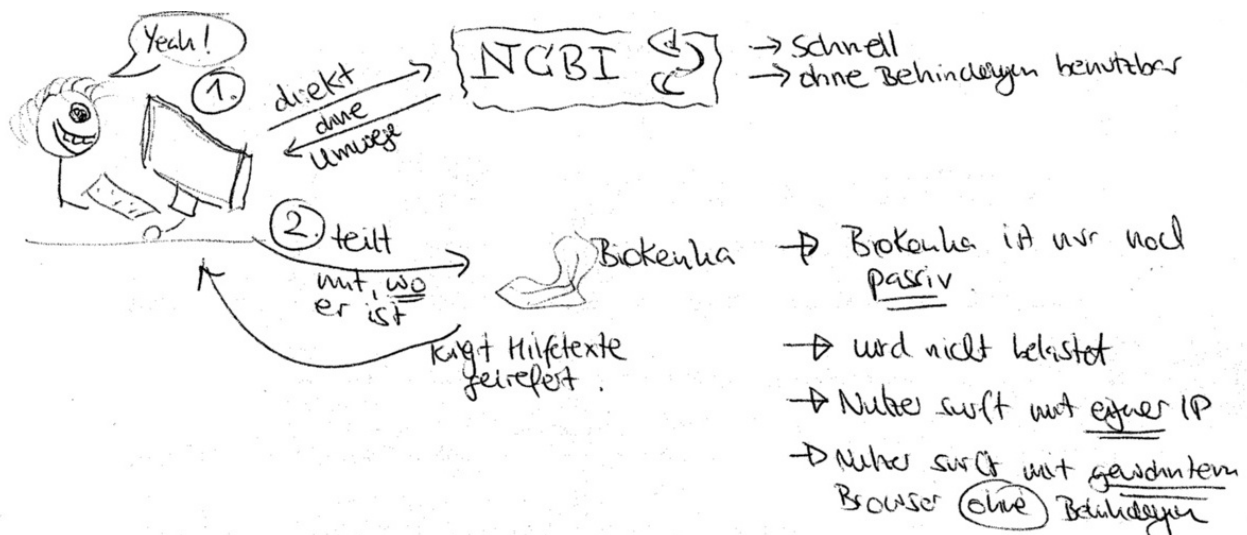


Abbildung 1: Funktionsskizze der neuen MetaSearch

Die neue MetaSearch setzt direkt am Browser des Benutzers an, etwa als eigenständiger Browser oder Browser-Erweiterung (Firefox-Extensions sind sehr populär und einfach zu installieren). Der Benutzer surft *direkt* und ohne Umwegen auf der Seite, wo er hinmöchte. Im Browser arbeitet nun die neue MetaSearch, die dann dynamisch mit dem BioKemika-Server kommuniziert und von ihm neue Hilfetexte bekommt, die sie anzeigt. Die neuen Grundprinzipien sind damit:

- Passivität: BioKemika spielt im Datentransfer nur noch eine passive Zuhörerrolle
- Integrativität: Der Benutzer kann mit seinem gewohnten Firefox-Browser surfen und die Extension nach Belieben an- und abschalten.
- Neutralität: Die Nutzdaten fließen nicht mehr über BioKemika, es können keine rechtlichen Probleme wegen IP-Adressen entstehen.

Das neue System ist offensichtlich zweigeteilt: Zum einen gibt es die Browser-Komponente, zum anderen das System auf dem BioKemika-Server, welches die Hilfetexte liefert. Offensichtlich lässt sich der Server-Teil austauschen, womit das Problem auf einen Schlag völlig portierbar wird. Oder, um es mit den Worten von Marius Schmidt zu sagen, die ironischerweise identisch mit meinen waren:

Dieses Konzept lässt sich beliebig adaptieren – ein Internet-Assistent, der Grundschkinder genauso durch das Internet führen könnte wie Senioren. “

”

Nun tun sich viele Fragen und Optionen auf. Sie sollen im Folgenden etwas sortiert werden.

Die Browser-Komponente

Welche Plattform?

Die Programmieraufwands-Kurve besitzt bei zwei Möglichkeiten ein lokales Minimum:

- MetaSearch als eigener **Browser**, auf Qt/Webkit basierend
- MetaSearch als Firefox-**Plugin**

Die erste Möglichkeit stellt eine eigene Anwendung dar, quasi einen eigenen selbstständigen Browser, den der Benutzer nutzt, wenn er MetaSearch benutzen möchte. Vorteil ist die Unabhängigkeit von anderen Browsern, dafür muss der Nutzer aber eine richtige eigene Software installieren, die man dann auch auf mehrere Plattformen portieren muss (Windows, Linux, Mac OS X) und warten muss. Dank Qt/Webkit ist das aber verhältnismäßig wenig Aufwand.

Die zweite Möglichkeit basiert auf dem populären Browser Mozilla Firefox, der meiner Empfindung nach an der Uni Frankfurt eine unheimliche Popularität genießt. Dieses Empfinden basiert darauf, dass quasi 90% der Studenten mit Notebook, die ich bis dato in Vorlesungshörsälen gesehen habe, mit Firefox surfen. Firefox gibt es für alle verbreiteten Plattformen. Eine herausragende Eigenschaft ist die exzellente Erweiterbarkeit dank XUL. Es gibt bereits viele Web 2.0-Anwendungen, die mit Firefox-Erweiterungen funktionieren. Ein unheimlicher Vorteil dieser Methode ist es, dass Firefox-Anwender ihr Browser-Umfeld nicht wechseln müssen – sie können die Extension einfach installieren und weitermachen wie bisher. Dieser nahtlose Übergang zwischen „mit“ und „ohne MetaSearch surfen“ ist mir eigentlich sehr wichtig. Aus diesem Grund würde ich wirklich sehr zu einer Firefox-Extension tendieren.

Funktionsweise & Aussehen

ObdA gilt, der Benutzer surfe zu www.ncbi.org. Bei einkommender HTTP-Antwort nimmt das Plugin die notwendigen Daten, stellt eine Anfrage an BioKemika und bekommt eine Website eine Website (also HTML!) zurück. Den stellt sie in einem Frame neben oder über dem Inhalt dar. Das Aussehen kann man sich noch überlegen, man könnte das Browserfenster einfach horizontal oder vertikal teilen (wie ein klassisches komplettes Frameset) oder das Inhaltsfenster wie in der alten MetaSearch in so einer Art IFrame darstellen. Mit XUL hat man (hoffe ich) sogar die Freiheit, das Aussehen komplett dynamisch/austauschbar zu gestalten.

Kommunikation mit BioKemika

Zur Kommunikation denke ich an so etwas wie REST bzw. SOAP. In dem eine Schnittstelle vorgeschrieben wird, ist damit schon die komplette Kommunikation definiert.

Von der Programmlogik hängt jetzt ab, welche Informationen beim Surfen alle übertragen werden sollen. Sicherlich die URL, wo der Benutzer hingeht/herkommt, POST und GET-Daten. Man könnte gegebenenfalls sogar den kompletten HTML-Text, den man erhalten hat, hochschicken (wobei man hier die Ladezeiten nicht vernachlässigen darf – Upload bei DSL ist meist langsam!). Will man später wirklich Dynamik in die Sache bringen, kann man auch Mauszeigerbewegungen und ähnliches reinbringen, um etwa Hilfestellungen zu geben, falls der Benutzer stockt (in der Annahme, dass er nicht weiterweiß).

Austauschbarkeit des Servers

Die Grundidee dieses Abschnittes basiert auf der, die wir alle von der berühmten Firefox-Extension „Adblock“ gewohnt sind. Kurzzusammenfassung: Sie blockiert Werbung auf Basis von Listen

(einfache txt-Dateien, sogenannte Filtersets), die irgendwo im Internet gespeichert sind. Man kann beliebige Listen zusammen arbeiten lassen. Bei der Installation der Extension öffnet sich ein Konfigurationsmenü, wo man einzelne Listen auswählen kann, die man gerne haben möchte.

Mit dem gleichen System soll die neue MetaSearch ausgerüstet werden, was sie gleichsam von BioKemika einen Schritt weit trennt. Statt sich an einen einzigen Server zu binden, wird er austauschbar gemacht (es gibt ja ein vorgeschriebenes Protokoll, mit dem die Serverkommunikation funktioniert). Welchen Server man auswählen will, das kann man nun in einer Einstellung vornehmen. So könnte man dann eigentlich auch mehrere Server gleichzeitig wählen. Letztlich wird BioKemika prinzipiell einer von vielen, die die Hilfetexte zur Verfügung stellen.

Nun könnten verschiedene Server verschiedene GUIs zur Verfügung stellen. Auf diese Weise könnte die Adaptierung für das Grundschulkind („Kinderguide im Internet“) etwa rote Rahmen um den kompletten Browser anzeigen, wenn der Benutzer auf böse Seiten im Internet kommt.

Die Serverkomponente

Wenn man den Browser-Teil gelesen hat, könnte man meinen, der Serverteil wäre relativ unspektakulär, weil er ja nur Texte liefert und so. Allerdings hat der Serverteil eine Doppelfunktion: Er liefert nicht nur Texte (und mit ihnen das Layout), sondern muss auch eine Möglichkeit geben, neue Texte einzugeben.

Prinzipiell implementieren wir hier ja Methoden des „Mitmachwebs“. Viele solcher Dienste haben einprägsame Homepages mit modernen AJAX-GUIs. Mir schwebt hier eigentlich vor, dass man auf die Homepage eines MetaSearch-Anbieters gehen könnte und dort auf eine Website kommt, wo man in strukturierter Weise eine Übersicht kriegt, was für Hilfetexte dort hinterlegt sind. Man soll dort Hilfetexte bearbeiten können, taggen, und was man halt sonst so macht. Andererseits soll die Bearbeitung der Hilfetexte auch direkt aus der Extension heraus gehen (was natürlich die Extension selbst nicht tangiert, da das ja auch im HTML läuft, was der BioKemika-Server etwa zurückgibt).

Plattform: Warum nicht als BioKemika-Anhängsel?

Ich will mich liebend gerne von MediaWiki trennen. Weil ich einfach kein PHP mehr schreiben will und das ganze sowieso nicht *in* MediaWiki arbeiten würde, sondern maximal als Anhängsel, so ähnlich, wie *proxy.php* bislang in der alten MetaSearch als eigenständiges Frontend gearbeitet hat, seine Daten aber aus der BioKemika-MetaSearch bekommen hat.

Der Vorteil, ein solches Anhängsel zu schreiben, wäre, dass man eben auf die BioKemika-Daten bequem zugreifen kann. Benutzer können im BioKemika-Umfeld schreiben und MediaWiki-Layout zum Formatieren der Hilfetextausgaben benutzen.

Allerdings sind diese Scheinvorteile leicht zu kompensieren: Die gemeinsame Datenquelle ist kein Vorteil, da jetzt schon MetaSearch ein Gefrickel mit Quasi-Textdateien im MediaWiki-Namensraum war, die bei jedem Auslesen geparst werden müssen. Damit sind die Möglichkeiten einer relationalen Datenbank nicht mal im Ansatz ausgereizt.

Die graphische Umgebung kann man außerdem völlig problemlos der BioKemika anpassen.

Plattform: Eine eigene Plattform muss her

Eine eigene Plattform mit eigenem Backend (Datenbank, etc.) in Programmiersprachen wie Python, Perl oder Java, muss geschrieben werden. Diese muss zum einen das REST-API bieten, um mit der Extension zu kommunizieren. Aber schon dort sollen die Möglichkeiten zum Bearbeiten von

Hilfetexten existieren, was direkt in die eigentliche Aufgabe der Darstellung einer Website entspricht. Eine genaue Vorstellung, wie diese Website aussehen soll, hab ich noch nicht, aber auf jeden Fall geht es primär um Übersichtlichkeit beim Verarbeiten von vielen Hilfetexten. Als Einflüsse werde ich mir einschlägige „Social web“-Anwendungen anschauen, wo auch oft große Datensätze mit Informationen über Websites verarbeitet werden.

[to be continued]