

Gnogito [ergo sum] Lastenheft

Autor: Matthias Kleine <kleine_matthias@gmx.de>

Letzte Änderung:

Version:

Historie: Siehe CVS-Log.

Repository: <http://cvs.berlios.de/cgi-bin/viewcvs.cgi/gnogito/gnogito/analyse>

Dateiname: lastenheft.sxw

Export: lastenheft.doc – Dieses Dokument wird unregelmäßig aus dem Original lastenheft.sxw exportiert. Im Zweifel ist letzteres immer auf einem aktuelleren Stand.

Status:

Lizenz: Dieses Dokument unterliegt der GPL.

Todo:

- Idee: Uhr für Lernsitzungen wird von Plattform bereitgestellt
- welche weiteren Features dieser Art kann man sich für eine Lernplattform vorstellen?
- Speicherung von Benutzerinformationen, auch bezogen auf Lerninhalte
 - das muss nun wirklich nicht jede Komponente selbst machen ...
- Unterscheidung: Was will der spätere Benutzer, was will der spätere Entwickler? Man kann sich das wie die Entwicklung einer Bibliothek vorstellen

Inhaltsverzeichnis

1 Was muss eine Lernplattform bereitstellen?	1
1.1 Erläuterung des Plattform-Gedankens	1
1.1.1 Einige Beispiele vorab	2
1.1.1.1 Lern-Hilfsmittel	2
1.1.1.2 Unterschiedliche Formen von Information	2
1.1.1.3 Organisation des Lernens	2
1.1.2 Integration lernbezogener Prozesse	2
1.1.3 Flexibles Erweiterungskonzept	2
1.1.4 Was macht eine Lernplattform aus?	3
1.1.4.1 Gemeinsamkeiten der Komponenten	3
1.1.4.1.1 Divide and Conquer	3
1.1.4.1.1.1 Ebene 6	3
1.1.4.1.1.1.1 Ebene 7	3
1.1.4.1.1.1.1.1 Ebene 8	3
1.1.4.2 Kommunikation zwischen den Komponenten	3

1 Was muss eine Lernplattform bereitstellen?

1.1 Erläuterung des Plattform-Gedankens

Ein zentraler Grundgedanke für Gnogito ist die Idee, eine Plattform für Lernkomponenten bereitzustellen. Was ist damit gemeint?

1.1.1 Einige Beispiele vorab

1.1.1.1 Lern-Hilfsmittel

Als Lern-Hilfsmittel kommen sehr unterschiedliche Werkzeuge in Betracht. Lernen kann beispielsweise durch Karteikarten, durch Mindmaps, durch Zusammenfassungen und vieles mehr unterstützt werden. Wissen kann mittels Tests geprüft oder gefestigt werden.

1.1.1.2 Unterschiedliche Formen von Information

Auch die Eingabe von Lerninformation kann sehr unterschiedliche Formen annehmen. Texte können formatiert werden, mathematische Formeln enthalten, mit Bildern illustriert werden, ja vielleicht möchte man sogar unmittelbar in einem Text grafische Illustrationen vornehmen können: Unterstreichungen malen, Pfeile oder Querverbindungen einzeichnen oder andere Aktionen durchführen, die beim Lernen mit gedrucktem Textmaterial durchaus üblich sind. Über Text hinaus ist denkbar, das Lernen auch mit multimedialem Material zu unterstützen: Von Ton- oder Videodateien über Flash-Animationen bis hin zu kleinen Lehrprogrammen (z.B. ein Generator für mathematische Rechenaufgaben u.ä.) ist hier vieles denkbar.

1.1.1.3 Organisation des Lernens

Lernen will auch organisiert sein: Für ein Studium erstellt sich ein Student idealerweise einen Studienplan, verwaltet Anmelde- und Prüfungstermine, führt Buch über Veranstaltungen, Scheine usw. Aber auch wer (noch) nicht (mehr) studiert möchte seine Lernprozesse womöglich organisieren: Der Berufstätige möchte z.B. seine Kenntnisse in einem Fachgebiet vertiefen und sich hierzu einen Meilensteinplan erstellen. Oder er möchte alte Kenntnisse in bestimmten Intervallen auffrischen und daran automatisiert erinnert werden. Bei derartigen organisatorischen Prozessen kann ebenfalls ein Programm behilflich sein.

1.1.2 Integration lernbezogener Prozesse

Die Grundidee von Gnogito besteht darin, lernbezogene Prozesse wie die oben skizzierten in einer einzigen Lernumgebung zu integrieren. Dabei ist bislang noch nicht klar, wie weit diese Integration überhaupt gehen soll. Sicherlich braucht nicht jede beliebige Aufgabe, die mit dem Lernen verbunden ist, innerhalb einer bestimmten Lernumgebung geleistet zu werden. So dürfte Lernen heute sicherlich mit einer Nutzung des Internet verbunden sein, es wäre jedoch kaum sinnvoll, für eine Lernumgebung Werkzeuge wie Browser oder Newsreader neu zu implementieren. Dies ist auch gar nicht zu leisten.

1.1.3 Flexibles Erweiterungskonzept

Gerade die Schwierigkeit, eine sinnvolle Grenze für konkrete lernbezogene

Werkzeuge zu ziehen, die innerhalb einer solchen Lernumgebung überhaupt Sinn machen, legt die Verwendung eines flexiblen Erweiterungskonzeptes nahe. Das technische Grundgerüst für dieses Konzept besteht in der Rich Client Platform von Eclipse 3.x. Das zugrundeliegende Plugin-Konzept ermöglicht eine aus modulare und damit sehr flexible Erweiterung. Davon ist jedoch nur die technische Sicht berührt. Eclipse 3.x stellt gewissermaßen eine flexible Laufzeitplattform bereit, aber was macht nun eine **Lernplattform** aus?

1.1.4 Was macht eine Lernplattform aus?

Die oben angeführten Beispiele haben die mögliche Vielfalt und Unterschiedlichkeit von lernbezogenen Werkzeugen aufgezeigt. Um aber mit Recht von einer integrierenden Plattform reden zu können, müssen diese Komponenten auch etwas Gemeinsames haben. Zudem sollen die Komponenten nicht losgelöst voneinander innerhalb einer scheinbar gemeinsamen Oberfläche existieren, sondern sie sollen miteinander kommunizieren, sich gegenseitig aufrufen, Informationen geben, Information anfragen, Information transformieren und vieles mehr. Was also haben die Komponenten gemeinsam, und in welcher Weise sollten sie miteinander kommunizieren können?

1.1.4.1 Gemeinsamkeiten der Komponenten

Die grundlegende Gemeinsamkeit besteht sicherlich in der Lernbezogenheit aller Komponenten. In welcher Weise lässt sich diese Gemeinsamkeit aber für softwaretechnische Zwecke ausnutzen?

1.1.4.1.1 Divide and Conquer

Lernbezogenheit ist ein zu grobes Kriterium. Wir müssen also feststellen, auf welche Arten eine Komponente sich für eine Lernapplikation eignen kann. Teilen wir also das Problem in kleinere Teilprobleme auf ...

1.1.4.1.1.1 Lerninhalte

Die meisten Komponenten dürften in irgendeiner Weise mit konkreten Lerninhalten zu tun haben. Betrachten wir also, in welcher Weise Komponenten mit konkreten Lerninhalten arbeiten können

1.1.4.1.1.2 Strukturieren von Lerninhalten

Lerninhalte müssen strukturiert sein. So sind sie z.B. sicherlich thematisch zusammenhängend angeordnet und könnten eine Hierarchie von über- und untergeordneten Themen bilden. Andererseits muss es möglich sein, die Inhalte sequentiell in einer sinnvollen Reihenfolge zu lernen. Um derartige Strukturen auf den Inhalten definieren zu können, benötigt man Werkzeuge, die eine Strukturansicht auf den Inhalten ermöglichen und eine Modifikation der Strukturen erlauben.

1.1.4.1.1.3 Eingabe von Lerninhalten

Lerninhalte müssen eingegeben werden. Man benötigt also „Editoren“ im weitesten Sinne.

1.1.4.1.1.4 Lernen von Lerninhalten

Lerninhalte müssen präsentiert werden. Sicherlich wird sich die Editieransicht

auf einen Inhalt von der Ansicht unterscheiden, die zum Lernen verwendet wird. Beim Lernen könnte z.B. eine Uhr für eine „Lernsitzung“ mitlaufen usw.

1.1.4.1.1.5 Prüfen von Lerninhalten

Gelernte Inhalte sollten idealerweise auch geprüft werden, um den Lernprozess einer Kontrolle zu unterziehen und um schwierig zu lernende Inhalte ggf. systematisch wiederholen zu können. Hierzu muss es einen Prüfmechanismus geben.

1.1.4.1.1.6 Metainformation zu Lerninhalten

Zu Lerninhalten können Benutzerinformationen gespeichert werden wie z.B. eine Versions- und Autorenhistorie, die Anzahl der Lernwiederholungen für diesen Inhalt, bestandene oder nicht bestandene Tests usw.

1.1.4.2 Kommunikation zwischen den Komponenten

1.1.4.2.1.1.1 Ebene 7

1.1.4.2.1.1.1.1 Ebene 8