Dezentrale Systementwicklung am Beispiel GNU/Linux

2. Meilenstein: Relevante Projekte

Projekt „MumieNav“
Entwicklung eines Navigationsnetzes für ein Lerntool

Michael Glässel – glaessel@cs.tu-berlin.de
Jörg Küster – kuester@cs.tu-berlin.de
Matthias Erche – erche@cs.tu-berlin.de

29.05.2002

Inhaltsverzeichnis

1.Einleitung

Für die Entwicklung von Software ist es wichtig zu überprüfen, ob man sich durch Wiederverwendung bereits bestehender Projekte Arbeit ersparen kann. Gerade die Bereiche dezentrale Softwareentwicklung und Open Source bieten weitreichende Möglichkeiten der Wiederverwendung von Software. Eine Betrachtung des Projektes MumieNav hat ergeben, dass der wohl einzige Ansatzpunkt für die Integration schon bestehender Software, ein eventuell eingesetzter XML- Parser darstellt. Wie beim ersten Meilenstein schon erwähnt, soll sich das Applet die darzustellenden Daten vom Server in Form eines XML- Datenstroms holen. Dieser muss clientseitig durch einen Parser ausgewertet werden. Da die Auswertung von einem Applet durchgeführt werden soll, ist es erforderlich, einen schnellen, Ressourcen-sparenden XML- Parser zu finden. Zusätzlich muss dieser unter einer Lizenz vertrieben werden, die kompatibel zu der des Mumie- Projektes ist.

2.JAXP

http://java.sun.com/xml

Die „Java API for XML Processing“ (JAXP) ist eine von SUN entwickelte API zum Verarbeiten von XML Dokumenten. Dabei kommen verschiedene Parser zum Einsatz. Standardmäßig ist die JAXP ab JDK 1.4 enthalten und es werden als Verarbeitungsansätze SAX und DOM angeboten.

1. SAX

Bei der Verarbeitung von XML- Dokumenten mit SAX wird ein ereignisorientierter Ansatz gewählt. Das Dokument wird in Stücken geladen und immer dann, wenn ein angemeldetes Element beim Parser vorbeikommt, meldet er dies in Form eines Ereignisses, das für die Verarbeitung abgefangen werden kann. Das hat den Vorteil, dass nicht das gesamte Dokument im Speicher gehalten werden muss. Der Nachteil liegt aber eindeutig im nicht wahlfreien Zugriff auf ein Element.

2. DOM

Das „Document Object Model“ (DOM) ist eine Entwicklung des W3C. Das Standard DOM wurde unabhängig von einer Programmiersprache konzipiert. Es handelt sich hierbei um einen Ansatz, bei dem man eine streng hierarchische Repräsentation des Dokuments erhält. Der Vorteil liegt hierbei im wahlweisen Zugriff auf ein Element. Allerdings hat dies im Vergleich zu SAX auch Effizienzprobleme zur Folge, da das gesamte Dokument im Speicher gehalten werden muss.

3.JDOM

http://www.jdom.org

„JDOM ist eine einfache Möglichkeit, XML-Dokumente leicht und effizient mit einer schönen Java-API zu nutzen. Im Gegensatz zu SAX und DOM, die unabhängig von einer Programmiersprache sind, wurde JDOM speziell für die Programmiersprache Java entwickelt. Durch die Optimierung der API auf Java ist eine wesentlich bessere

Performance und eine bessere Speichernutzung als bei DOM möglich."[1]

Das Dokumentenmodell von JDOM liegt eine Ebene über dem von DOM und SAX. Eine Entwicklung mit DOM oder SAX würde eine späteres Ändern des Modells verhindern, da sich die Programmiermodelle voneinander unterscheiden. Mit JDOM stellt das kein Problem dar, da es einerseits die Vorteile beider Modelle vereinigt und Schnittstellen zu beiden Modellen bietet. Aktuelle (und zukünftige) Standards wie DOM Level 2, SAX 2.0 oder XML- Schema werden dabei unterstützt.

Es wird ebenfalls eine interne Repräsentation des Dokuments erstellt, die der von DOM ähnlich ist, allerdings nutzt diese Java- spezifische Datenstrukturen, die eine effiziente Speicherung und einen optimalen Zugriff ermöglichen.

JDOM wurde von SUN als ein Java Specification Request (JSR-102) akzeptiert. Dies ist wohl nicht üblich für eine API, die eine schon enthaltene Funktionalität reimplementiert aber SUN soll wohl selbst der Meinung sein, JDOM sei einfacher zu benutzen, als eine frühere API[2].

4.Sourceforge Projekte

Eine gute Quelle, um Software für die Wiederverwendung zu finden, ist das Sourceforge Projekt[3]. Eine Recherche hat ein paar relevante Projekte ergeben, die hier aber nur kurz Erwähnung finden sollen:

- XMLtp http://sourceforge.net/projects/jxmltp:

nichtvalidierender Parser, der für den serverseitigen Einsatz entwickelt wurde

- Xparse-J http://sourceforge.net/projects/xparse-j

„Xparse-J aspires to be the smallest Java XML parser on the planet."[4]

- NanoXML http://nanoxml.sourceforge.net/

nichtvalidierender Parser

- Piccolo http://sourceforge.net/projects/piccolo

nichtvalidierender Parser mit JAXP Unterstützung

5.Fazit

Die Verwendung der bei Sourceforge gefundenen Projekte wurde kaum in Betracht gezogen. Es ist wahrscheinlich, dass diese Lösungen im Vergleich zu den vorher vorgestellten APIs ein schnelles, schlankes Ergebnis produzieren, doch andere

1 aus „Java ist auch eine Insel" von Christian Ullenboom
2 aus dem Vortrag „JDOM Makes XML Easy" von Jason Hunter auf der JavaOne 2002
3 www.sorceforge.net
4 http://webreference.com/xml/tools/xparse-j.html

Punkte lassen Zweifel offen. So muss man bedenken, dass das Projekt Mumie noch lange weiterentwickelt werden wird. Es ist deshalb zwingend erforderlich, dass die verwendete Lösung kontinuierlich gepflegt und an neue Entwicklungen und Standards angepasst wird, um dem Gesamtprojekt dem Anspruch eines zeitgemäßen Systems gerecht zu werden. Bei den Sourceforge Projekten ist allerdings nicht abzusehen, wie lange sie noch bestehen werden.

Es stellt sich nun die Frage, welche XML- API für das MumieNav- Projekt geeignet ist.

DOM und SAX haben eine Gemeinsamkeit. Sie wurden beide unabhängig von einer Programmiersprache entwickelt und sind sehr generisch gehalten. Eine direkt für eine spezielle Sprache entwickelte API kann natürlich deren Eigenheiten nutzen, um eine optimierte Verarbeitung eines XML- Dokuments zu ermöglichen. Sowohl bei JAXP, als auch JDOM, ist es möglich, eigene oder die Standard Parser zu verwenden.

Im Moment steht noch nicht fest, welcher Ansatz verfolgt wird. Das liegt unter anderem an Unklarheiten, was die Lizenz des Gesamtprojektes angeht und einer genaueren Prüfung durch evtl. Testimplementierungen. Tendenziell lässt sich aber sagen, dass die JDOM API zur Entwicklung benutzt werden wird.