

Verbesserung der Handhabbarkeit von komplexen Softwaredokumentationen durch Visualisierung von Dokumentbeziehungen

Dan Klingenberg
klingenb@cs.uni-magdeburg.de

betreuender Lehrstuhl:
Prof. Dr. rer. nat. habil. Gunter Saake
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme
Postfach 4120, D-39016 Magdeburg, Germany

1 Einleitung und Motivation

Die Softwaredokumentation wird von vielen kleinen und mittleren Softwarefirmen immer noch sehr „stiefmütterlich“ behandelt, obwohl sie mittlerweile ein Erfolgskriterium geworden ist. Große Unternehmen, wie z.B. die Bayer AG, können es sich nicht mehr leisten, Programme und Programmabläufe undokumentiert zu erstellen. Daher wird immer mehr Aufwand und Geld in die lückenlose Softwaredokumentation investiert. Dieser Aufwand ist notwendig, um die hohen Anforderungen der Wirtschaftsprüfer oder der US Food and Drug Administration (FDA)(siehe <http://www.fda.gov>) der Pharmaindustrie erfüllen zu können.

Abgeleitet aus diesen hohen Ansprüchen an die Softwaredokumentation ist es unmöglich, dass die Dokumentation einer komplexen Software nur aus einfachen Quellcodekommentaren bestehen kann. Eine Softwaredokumentation muss ausführlich die Programmaufgabe, Programmabläufe und Programmteile beschreiben. Je umfangreicher die Dokumentation einer Software ist, desto komplizierter ist es für den Nutzer die komplexen Strukturen und Zusammenhänge innerhalb der Dokumentation zu verstehen. Der Grund hierfür liegt nicht allein im Umfang der Softwaredokumentation, es fehlen häufig geeignete Werkzeuge zur Darstellung von Softwaredokumentationen und deren Strukturen. Eine mögliche Lösung bietet sich in der Visualisierung von Dokumentbeziehungen. Diese Darstellungsmöglichkeit unterstützt den Benutzer, einfach und schnell komplexe Dokumentationsstrukturen aufzunehmen, zu verstehen und Fehler visuell zu erkennen.

Ziel der Diplomarbeit [Kli02], welche diesem Kurzbeitrag zu Grunde liegt, war es, ein Konzept zu entwerfen, das in der Lage ist, Informationen über Softwaredokumentationen aus verschiedensten Systemen (Datenbank Management Systeme, Filesysteme) zu extrahieren, visuell und übersichtlich aufzubereiten.

2 Anforderungen an die Visualisierung von Dokumentbeziehungen

Dokumentbeziehungen stellen in diesem Zusammenhang eine Menge von Informationen dar, die auf verschiedenste Art und Weise visualisiert werden können. Die Fachliteratur bietet hierfür ein reiches Spektrum an unterschiedlichen Konzepten und Techniken [Mun97].

Um eine Entwicklung eines geeigneten Systemkonzepts sowie einer Visualisierungstechnik zu ermöglichen, sind im ersten Schritt die technischen und fachlichen Anforderungen zu definieren. Im Gespräch mit mehreren Dokumentaren konnten die folgenden wichtigen fachliche Anforderungen bestimmt werden:

- Die Bedienung und Navigation ist für den Benutzer möglichst einfach zu halten. Da die Softwaredokumentation eine sehr große Anzahl von Dokumenten umfassen kann, ist es erforderlich dem Nutzer zusätzliche Funktionen zur Navigation und Interaktion zur Verfügung zu stellen. Diese Funktionen können Zoom, Translation oder Rotation umfassen.
- Die Abbildung der Softwaredokumentation muss vollständig erfolgen. Zu diesem Zweck ist es nötig, das Dokumentnetzwerk so anzuordnen, dass der Nutzer einfach und schnell semantische und strukturelle Zusammenhänge erkennen kann. Weiterhin ist es wichtig, Techniken einzusetzen, die es erlauben, Teile des Graphen komprimiert bzw. detailliert darzustellen, ohne dass der strukturelle Zusammenhang verloren geht.

Im Hinblick auf die prototypische Umsetzung des Konzeptes sind zwei wesentliche technische Anforderungen zu berücksichtigen:

- Eine Vielzahl der heutigen Softwaredokumentationen werden in Datenbanken oder Filesystemen abgelegt und gespeichert. Dies erfolgt nicht nach einem einheitlichem Datenbankschema bzw. Muster, sondern jedes Unternehmen verfolgt dort seine eigene Strategie. Somit besteht die Aufgabe des Prototypen darin, eine Schnittstelle, die sich dynamisch anpasst, zur Verfügung zu stellen. Hierfür muss ein Modul geschaffen werden, welches einen direkten Zugriff auf unterschiedliche Datenbanken, aber auch auf statische Text-Dateien erlaubt. Als Austauschformat eignet sich hier XML (Extensible Markup Language) [Wor02] besonders gut.
- Auf Grund des großen Umfangs der darzustellenden graphischen Elemente (Dokumentbeziehungen) ist der Einsatz einer 3D-Graphikbibliothek sehr hilfreich. Zum einen bieten diese Bibliotheken eine Vielzahl von verschiedenen Funktionen, die die Umsetzung stark vereinfachen, zum anderen nutzen moderne 3D-Graphikbibliotheken direkt Funktionen der Graphikkarte, welche die Anwendung um ein Vielfaches beschleunigen.

3 Gesamtarchitektur

Die Gesamtarchitektur des Prototypen basiert auf einem modularen Ansatz. Dies ist erforderlich, um eine einfache Anpassung an verschiedene Datenquellen sowie unterschiedlichen Darstellungsvarianten zu ermöglichen. Die drei Module der Gesamtarchitektur (*Schnittstellen-Modul*, *Datenhaltung-Modul*, *Visualisierung-Modul*) sind hierarchisch angeordnet und bestehen wiederum aus einzelnen Komponenten. Ein Modul stellt in diesem Zusammenhang eine logische Gruppe von Funktionen und Aufgabenbereichen dar.

Die primäre Verbindung zwischen der graphischen Darstellung und einem Fremdsystem, in dem die Softwaredokumentation gespeichert ist, bildet das **Schnittstellen-Modul**. Es umfasst wiederum einzelne Komponenten, die speziell für die Kommunikation mit einem Fremdsystem entworfen werden. Der Zugriff auf ein Fremdsystem kann direkt über eine Verbindung zur Datenbank oder über eine statische *XML-Datei*, in die die Daten zuvor exportiert wurden, erfolgen. Das **Datenhaltung-Modul** wird durch die Datenhaltungskomponente repräsentiert und hat folgende Aufgabe: es verwaltet und speichert die Daten, die durch das Schnittstellen-Modul

bereit gestellt werden und bereitet sie für das Visualisierung-Modul auf. Den Kern der Gesamtarchitektur stellt das **Visualisierung-Modul** dar. Dieses Modul hat die Aufgabe, die Daten graphisch ansprechend und übersichtlich dem Nutzer zu präsentieren. Weiterhin stellt es eine Nutzerschnittstelle zu Verfügung, über die der Nutzer mit dem System in Interaktion treten kann. Ausgehend von diesen Aufgabenbereichen ist das Visualisierung-Modul in drei Komponenten aufgeteilt (*Layoutkomponente*, *Adaptive Technik-Komponente*, *Darstellungs- und Interaktionskomponente*).

Die Layoutkomponente implementiert einen Layoutalgorithmus, der den gerichteten Graphen nach vorgegebenen Kriterien, z.B. Minimierung der Kantenüberschneidungen, anordnet. Für die konkrete Implementierung des Prototypen kommt ein *FADE* Algorithmus [Qui01] zum Einsatz. Auf Grund des sehr großen Umfangs einer Softwaredokumentation, ist es erforderlich, eine weitere Komponente (Adaptive Technik-Komponente) einzufügen, die eine Visualisierungstechnik zur Verbesserung der übersichtlichen Gestaltung des Graphen implementiert. Eine geeignete Technik bietet sich hier mit dem *Fish Eye View* [Fur81]. Die graphische Visualisierung auf dem Bildschirm und die Interaktion mit dem Nutzer erfolgt durch die Darstellungs- und Interaktionskomponente.

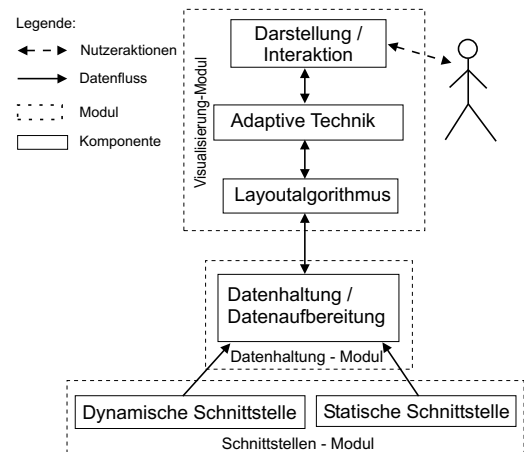


Abbildung 1: Gesamtarchitektur des Prototypen

4 Zusammenfassung und Ausblick

Angesichts der begrenzten Zeit und des weit gefassten Themenbereiches konnte eine vollständige Implementierung des Entwurfs und eine genaue Analyse des Laufzeitverhaltens nicht vorgenommen werden. Zu Beginn dieser Arbeit lag der Fokus auf die Entwicklung eines Prototypen für eine spezielle Softwaredokumentation. Dieser verlagerte sich auf Grund der vielfältigen Visualisierungsmöglichkeiten zum Entwurf einer modularen Gesamtarchitektur, die für die Visualisierung verschiedener Softwaredokumentationen eingesetzt werden kann. Das nächste Ziel ist somit die vollständige Implementierung des Entwurfs und die ausführliche Untersuchung des Laufzeitverhaltens sowie der Test auf Akzeptanz beim Nutzer.

Literaturverzeichnis

- [Fur81] Furnas, G. W.: The FISHEYE View: A New Look at Structured Files. Technical report, AT&T Bell Laboratories, 1981.
- [Kli02] Klingenberg, D.: Verbesserung der Handhabbarkeit von komplexen Softwaredokumentationen durch Visualisierung von Dokumentbeziehungen., 2002.
- [Mun97] Munzner, T.: H3: Laying out large directed graphs in 3D hyperbolic space. In Lavagno, L.; Reisig, W. (Hrsg.): *Proc. IEEE Symp. Information Visualization*, S. 2–10, 20–21 Oktober 1997.
- [Qui01] Quigley, A. J.: *Large Scale Relational Information, Visualisation, Clustering and Abstraction*. Dissertation, University of Newcastle, 2001.
- [Wor02] World Wide Web Consortium (W₃C): *Extensible Markup Language (XML), Version 1.1*, 2002. <http://www.w3.org/TR/xml11/>.