

# ANPASSBARE, EXPLORIERBARE VIRTUELLE INFORMATIONSRÄUME ZUR QUALITÄTSBEWERTUNG GROSSER SOFTWARE-SYSTEME

- Erste Erfahrungen -

Frank Simon, Frank Steinbrückner, Claus Lewerentz  
Software Systems Engineering Research Group  
Technical University Cottbus, Germany  
E-mail: (simon | fsteinbr | cl)@informatik.tu-cottbus.de

## 1 Einleitung

Die Qualitätsbewertung großer Softwaresysteme spielt im Vorfeld des Reengineering eine wichtige Rolle, da hiermit - nach entsprechender Konkretisierung des verwendeten Qualitätsbegriffs mittels entsprechender Techniken wie z.B. GQM [BaWe84] - ein präzises Bild über den jeweiligen Grad der Erfüllung der an ein Produkt gestellten Qualitätsanforderungen entsteht. Dieses kann z.B. innerhalb der Entscheidungsmatrix von Jacobson [JaLi91] (und in der erweiterten Entscheidungsmatrix von Schmidt und Postema, [ScPo98]) verwendet werden, um zu entscheiden, ob ein Produkt überhaupt überarbeitet werden soll oder ob eine Neuentwicklung aufgrund großer qualitativer Defizite zu bevorzugen ist. Im Falle der Entscheidung für das Reengineering kann dieses Bild ebenfalls für eine Abschätzung der Kosten für konkrete Reengineering-Aktivitäten verwendet werden (vgl. z.B. [StDu97]).

Um die Qualität von großen Softwaresystemen möglichst effizient beurteilen zu können, ist ein wesentliches Konzept der Betrachtung die *Abstraktion*, d.h. das zielgerichtete Hervorheben der für die Überprüfung einer Anforderung relevanten Daten bei gleichzeitiger Reduktion dafür irrelevanter Daten. Für eine praxisrelevante Qualitätssicherung ist eine weitere wesentliche Anforderung die *Werkzeugunterstützung*. Bisherige Werkzeuge als Abstraktionen darstellende Unterstützung bei der Qualitätsbewertung großer Systeme haben folgende Probleme:

- Sie bieten meistens nur eine Abstraktionssicht (z.B. den Vererbungsbaum); im Falle mehrerer Abstraktionssichten werden diese orthogonal zueinander dargestellt, d.h. Zusammenhänge zwischen den Sichten bleiben verborgen (z.B. ein Call-Graph von Methoden und ein Vererbungsbaum). Grundsätzliche Vorgehen wie *Top-Down* oder *Bottom-Up* werden dadurch wesentlich behindert. Dieser Nachteil wiegt besonders bei der Qualitätsbewertung schwer, da diese i.d.R. von Personen durchgeführt wird, die das System (noch) nicht kennen. Eine Lösung dieses Problems ist z.B. in [JaGaRi99] anhand eines konkreten Werkzeugs beschrieben, in dem ein *Abstraktionskonzept*, das der Verfeinerung der Enthaltenseinsrelation, mehrfach angewendet wird und dadurch eine Hierarchiedarstellung ermöglicht. Die verschiedenen Abstraktionssichten gehen dabei ineinander über, was sowohl das Top-Down- als auch das Bottom-Up-Vorgehen unterstützt. Eine Erweiterung auf andere Abstraktionskonzepte (z.B. Vererbung, Benutzung, etc.) ist allerdings nicht möglich.
- Sie bieten nur fixe Abstraktionssichten an: Es ist weder möglich, konkrete Softwareteile während der Analyse auszublenden, noch zusätzliche Informationen einzublenden. Ebenfalls das für die Qualitätsbewertung sinnvolle Annotieren untersuchter Teile ist nicht möglich.
- Sie können in zwei, i.d.R. disjunkte Werkzeugklassen eingeordnet werden: *Meßwerkzeuge*, deren Abstraktionen lediglich aus numerischen Werten der betrachteten Systemteile bestehen, und *Strukturwerkzeuge*, die versuchen, relevante Objekte und Relationen zwischen ihnen möglichst aussagekräftig darzustellen. Ausnahmen sind z.B. *CodeCrawler*, der u.a. die Darstellungsparameter von Klassen (Breite, Höhe, Farbe) in Vererbungsäumen in Anlehnung an bestimmte Meßwerte festlegen kann [DeDuLa99], oder eine Erweiterung von UML-Klassendiagrammen durch meßwertbasierte Einfärbung [Przy99].
- Sie sind i.d.R. nicht in der Lage, dem jeweilig verwendeten Qualitätsmodell entsprechend angepaßt zu werden: Diese Anpassung ist speziell bei Meß-Werkzeugen aber sehr wichtig, damit die ermittelten Meßwerte im jeweiligen Kontext zielgerichtet interpretiert werden können.

- Sie bieten zu feine oder zu grobe Abstraktionen an:
  - Zu feine Abstraktionen skalieren nicht für praxisrelevante Größen von zu betrachtenden Softwaresystemen. So sind z.B. Vererbungsbäume und Klassendiagramme mit mehr als 100 Klassen nicht mehr verständlich darstellbar. Ebenfalls zeilenorientierte Visualisierungen scheitern bei größeren Systemen (in *Seesoft*<sup>®</sup> wird die Grenze z.B. bei 50KLOC angegeben, [EiStSu99]).
  - Zu grobe Abstraktionen skalieren zwar, abstrahieren aber so sehr vom System, daß keine Detailinformationen mehr gewonnen werden können (z.B. Verteilungsdiagramme von Meßwerten, vgl. [DeDuLa99] oder [SiRuLe00]).

Viele dieser Probleme treten grundsätzlich bei der verständlichen Darstellung großer Datenmengen auf. Ein wesentliches, in diesem als *Information Retrieval* bezeichneten Bereich bereits sehr erfolgreich eingesetztes Konzept ist die Erstellung sogenannter *virtueller Informationsräume*, die im Gegensatz zum Darstellen physischer Datenräume auf abstrakten Raum- und Objektstrukturen basieren ([DäPa98], S. 40ff). Diese werden z.B. bereits eingesetzt für die Exploration großer Webstrukturen und Datenbanken (z.B. *SemNet* in [FaPoFu99]; weitere Beispiele u.a. in [Chen99]).

In dieser Arbeit sollen erste Erfahrungen beim Einsatz eines am *Lehrstuhl Software-Systemtechnik* der *Technischen Universität Cottbus* entwickelten virtuellen Informationsraums zur Qualitätsbewertung großer objektorientierter Systeme vorgestellt werden. Für diesen Zweck wird in Abschnitt 2 der inhaltliche Datenraum (d.h. die grundlegenden, darzustellenden Daten) des virtuellen Informationsraums erschlossen und für die Darstellung vorbereitet. Im Abschnitt 3 wird ein Werkzeug zur Exploration eines darauf aufbauenden virtuellen Informationsraums vorgestellt, das die oben aufgezeigten Probleme löst und effizient für die Qualitätsbewertung eingesetzt werden kann. Ein erster Erfahrungsbericht ist in Abschnitt 4 gegeben: Dort ist ein Projekt beschrieben, in dem ein großes JAVA-Projekt mit dem vorgestellten Werkzeug qualitativ bewertet wird und diese Ergebnisse von den Entwicklern des Systems beurteilt werden. Die äußerst positiven Ergebnisse belegen den Wert dieses Ansatzes, der in Abschnitt 5 noch einmal zusammenfassend dargestellt wird.

## 2 Erstellung von virtuellen Informationsräumen zur Qualitätsbewertung

Die grundsätzlichen Schritte zur Erstellung eines virtuellen Informationsraums sind ([DäPa98], S. 55ff)

1. die inhaltliche Erschließung des Datenraums,
2. die quantitative Analyse der Objektähnlichkeiten und
3. die raumbezogene Darstellung der Objektrelationen.

Alle drei Schritte werden im folgenden jeweils kurz dargestellt:

### 2.1 Erschließung des Datenraums

Wesentliche Informationen, die für die Erstellung eines virtuellen Informationsraums zur Beurteilung großer objektorientierter Softwaresysteme relevant sein können und die für das in Abschnitt 4 beschriebene Projekt verwendet werden, sind:

- *Projekthierarchie*: Diese hierarchische Enthaltenseinsstruktur stellt jeweils alle Entitäten der Abstraktionsschichten Subsystem, Datei, Klasse und Methode/Attribut dar.
- *Strukturdaten*: Diese Daten bezeichnen konkrete Relationen zwischen betrachteten Objekten. Möglich sind z.B. aus dem Quelltext automatisch extrahierbare Relationen „Methode benutzt Methode“, „Methode benutzt Attribut“ oder „Klasse erbt von Klasse“. Die Relationen können jeweils auf alle übergeordnete Abstraktionsschichten hochaggregiert werden (z.B. Subsystem erbt von Subsystem).
- *Kohäsionsdaten*: Diese Daten geben Aufschluß darüber, welche betrachteten Elemente entsprechend gemachter Vorgaben zusammengehören. Diese Informationen sind wichtig für die Bildung neuer Abstraktionsschichten, die sich durch Gruppierung von betrachteten Elementen bilden. Möglich sind z.B. Vererbungskohäsion, die das Bilden von Vererbungsteilbäumen ermöglicht, Benutzungskohäsion, die das Bilden von Gruppen miteinander interagierender Klassen ermöglicht, oder allgemeine Merkmalskohäsion, die auf den gemeinsamen Besitz von Merkmalen beruht (z.B. auf dem Merkmal Autorenschaft, die das Zusammengehören von Dateien entsprechend ähnlicher Autoren betrachtet).
- *Objektdaten*: Diese Daten geben nähere Informationen zu einem betrachteten Objekt. Hierzu zählen besonders die verschiedenen Meßwerte, die für jedes Objekt ermittelt werden können. Darüber hinaus können hier während der Exploration ermittelte Ergebnisse als Annotationen hinzugefügt werden.

- *Systemdaten*: Die Daten geben nähere Informationen zum gesamten System. Dieses schließt sowohl globale Maße (z.B. Anzahl von Klassen) als auch Verteilungsdiagramme verschiedener Maße ein (z.B. Auskunft über Extremwerte und deren Häufigkeit bzw. Abstand zum durchschnittlichen Wert).

## 2.2 Quantitative Analyse der Objektähnlichkeiten

Dieser Schritt ist wichtig, um die Positionierung der betrachteten Elemente im virtuellen Informationsraum möglichst aussagenstark vornehmen zu können. Im Gegensatz zu klassischen Techniken der Graphvisualisierung, die größtenteils „technisch“ bedingte Positionen verwenden (z.B. zur Minimierung der Kantenüberschneidungen), sollen innerhalb des betrachteten Systems zusammengehörende Elemente auch innerhalb des virtuellen Informationsraums eng beieinander liegen. Die dafür notwendigen Objektähnlichkeiten sind in Anlehnung an die Kohäsionsdaten zu wählen. So kann die Vererbungskohäsion z.B. auf Objektähnlichkeiten bzgl. der Menge gemeinsamer Oberklassen abgebildet werden.

Mittels eines speziellen generischen Distanzmaßes kann dieser Schritt der Bestimmung von Objektähnlichkeiten jeweils einfach an konkrete Informationswünsche angepaßt werden ([SiLöLe99]).

## 2.3 Raumbezogene Darstellung der Objektrelationen

Die im vorigen Schritt ermittelten Ähnlichkeiten können auf Distanzen im euklidischen Raum transformiert werden (vgl. [Chen99]). Darüber hinaus können folgende Darstellungsparameter für die Darstellung der Informationen verwendet werden:

- Größe und Farbe können durch numerische Objektdaten bestimmt werden,
- Die Form kann jeweils in Anlehnung der jeweiligen Ebene eines Objektes innerhalb der Abstraktionshierarchie verwendet werden (z.B. eine Klasse als Kugel, eine Datei als Quader etc.),
- Zusätzliche Kanten können durch Darstellung von konkreten Strukturdaten verwendet werden. Die Richtungen der verschiedenen Relationen können durch unterschiedliche Farbverläufe dargestellt werden.

## 3 CrocoCosmos: Werkzeug zur Exploration anpaßbarer virtueller Informationsräume

CrocoCosmos ist ein Werkzeug zur vollautomatischen Erstellung anpaßbarer, virtueller Informationsräume für die Qualitätssicherung großer objektorientierter Systeme. Den zu erschließenden Datenraum bezieht CrocoCosmos größtenteils aus dem Meßprogramm Crocodile. Crocodile selbst ist ein für große objektorientierte Systeme einsetzbares Werkzeug zur Bestimmung von Meßwerten von frei definierbaren Maßen. Da alle durch Crocodile ermittelten Daten in ein Repository in Form einer relationalen Datenbank gefüllt werden, kann CrocoCosmos direkt auf diesem aufsetzen.

CrocoCosmos besitzt als primäres Interaktionsmedium eine Sicht auf den dreidimensionalen Informationsraum. Zur Beschreibung der Welt wird die Sprache VRML, für das Erzeugen der Sichten ein Standard-VRML-Betrachter verwendet. Um diese Welten dynamisch modifizieren zu können und gleichzeitig mit ihnen interagieren zu können, wird das EAI-Interface des VRML-97-Standards verwendet. Die Positionierung der Elemente kann durch definierbare Instanzierungen des generischen Distanzmaßes bestimmt werden ([SiLöLe99]), so daß bereits die relativen Positionen der dargestellten Elemente in einer Sicht auf den Informationsraum wertvolle, anpaßbare Informationen vermittelt. Für das beschriebene Projekt wurde damit die Vererbungskohäsion, die Methoden-Benutzungskohäsion und die Attributbenutzungskohäsion, die durch gemeinsam verwendete Attribute entsteht, berücksichtigt.

Innerhalb des dreidimensionalen Informationsraums sind folgende Techniken dynamisch anwendbar:

- *Abstraktionsniveauselektion*: Es kann sowohl für alle dargestellten Elemente als auch für einzelne Elemente das jeweils zu verwendende Abstraktionsniveau gewählt werden. So können z.B. durch das Anklicken eines Pakets alle seine Dateien angezeigt werden.
- *Navigation*: Es können verschiedene Sichten auf den Informationsraum erstellt werden. Standardtechniken wie Turn (zum Drehen um die Welt) oder Slide (zum Fliegen durch die Welt) ermöglichen eine einfache Lösung des *Focus-vs-Context-Problem* [Chen99]: Die Welt kann sowohl zur Gewinnung einer Gesamtübersicht, als auch für Detailanalysen verwendet werden.

- *Ein- /Ausblendung von Strukturdaten.* Es können sowohl für alle dargestellten Elemente als auch für einzelne Elemente unterschiedliche Relationen in Form zusätzlicher, gefärbter Kanten dargestellt werden.

Neben der dreidimensionalen Welt wird parallel dazu eine zweidimensionale, in Anlehnung an den Windows-Explorer gestaltete Projekthierarchie dargestellt, die konsistent zur 3D-Welt gehalten werden kann. Jede Verfeinerung einzelner Projektteile in der Projekthierarchie wird demnach in der 3D-Welt nachgeführt. Darüber hinaus können Systemdaten und nicht durch Darstellungsparameter darstellbare Objektdaten auf Wunsch durch Kiviatdiagramme und Barcharts dargestellt werden. Ein typischer Screenshot von CrocoCosmos ist in Anhang A dargestellt.

### 3.1 Beispiel der Anwendung für die Qualitätsanalyse objektorientierter Systeme: Kohäsion

Eine häufige Anforderung an Subsysteme ist die Kohäsion der enthaltenen Klassen zueinander, d.h. eine hohe Kopplung zwischen den Subsystemklassen und eine geringe, über eine kleine Menge von Klassen stattfindende Kopplung zu Klassen anderer Subsysteme. Diese Qualitätsanforderung kann innerhalb des von CrocoCosmos erstellten, virtuellen Informationsraums effizient überprüft werden: Durch die Methoden-Benutzungskohäsion werden Klassen, die miteinander oder mit einer ähnlichen Menge von Klassen interagieren eng beieinander liegend plaziert. Werden die Klassen innerhalb des virtuellen Informationsraums entsprechend ihrer Subsystemzugehörigkeit gefärbt, d.h. Klassen desselben Subsystems besitzen dieselbe Farbe, so bilden sich in entsprechend dieser Qualitätsanforderung guten Systemen Gruppen von Klassen gleicher Farbe. Die für die Kopplung zwischen beiden Gruppen verantwortlichen Klassen bilden die größte Nähe zwischen den beiden Gruppen. Durch die Einblendung der konkreten Strukturdaten kann darüber hinaus die Richtung der Kopplungen untersucht werden. Jede Abweichung innerhalb einer Sicht auf den Informationsraum von diesem Idealzustand ist ein guter Startpunkt für die Identifikation eines Qualitätsdefizits.

### 3.2 Beispiel der Anwendung für die Qualitätsanalyse objektorientierter Systeme: Vererbung

Eine häufiges Qualitätsdefizit objektorientierter Systeme ist der falsche Einsatz von Vererbung. Mit der im obigen Abschnitt beschriebenen Parametrisierung eines virtuellen Informationsraums können z.B. diejenigen Fälle identifiziert werden, bei denen die Vererbung besser durch eine Assoziation ersetzt werden sollte. Speziell Generalisierungs-/Spezialisierungsstrukturen weisen häufig einen ähnlichen Benutzungskontext auf, der in der Welt als Gruppierung von Klassen erkennbar ist. Verwendet eine Klasse nur eine Oberklasse, statt wirklich eine Spezialisierung darzustellen, so ist ihr Benutzungskontext ein anderer, d.h. ihre Position ist losgelöst von den anderen Klassen der betrachteten Vererbungsstrukturen. Nach Einblendung der konkreten Vererbungsstrukturen in die durch Methoden-Benutzungskohäsion positionierten Elemente ist eine derartige Besonderheit einfach zu identifizieren.

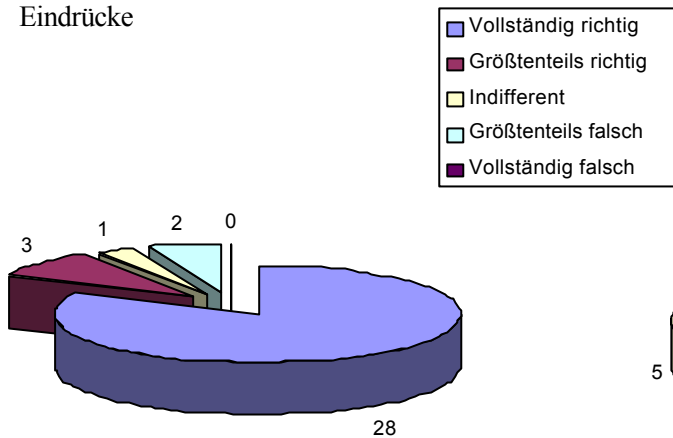
## 4 Erfahrungen beim Einsatz von CrocoCosmos: Eine Fallstudie

Das im vorigen Abschnitt vorgestellte Werkzeug wurde in einem Projekt zur Qualitätsbewertung eines JAVA-Systems mit 113 Paketen, 1182 Klassen, 10556 Methoden und 4717 Attributen verwendet. Der Auftraggeber hat das Projekt als Auftragssoftware erstellen lassen und verwendet die erstellte Qualitätsbewertung zur Abschätzung der Wartungskosten am System.

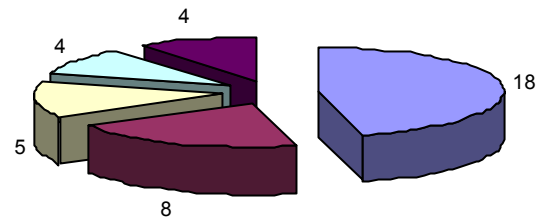
Die Qualitätsbewertung selbst wurde am Lehrstuhl Software-Systemtechnik durchgeführt und benötigte 25 Personenstunden. Das Ergebnis war ein Report mit 34 Eindrücken und 39 Restrukturierungsempfehlungen. Die Eindrücke bezogen sich dabei auf ein Qualitätsmodell, das im Vorfeld gemeinsam mit dem Auftraggeber erarbeitet wurde. Die Restrukturierungsempfehlungen umfassen negative qualitative Eindrücke und die für ihre Verbesserung vorgeschlagenen Aktionen.

Um die im Report gemachten Aussagen zu validieren, bestand die Möglichkeit, zwei Entwickler, die mit dem untersuchten System vertraut waren, die einzelnen Aussagen bzgl. der Validität beurteilen zu lassen. Das Ergebnis ist in der folgenden Abbildung zusammengefaßt:

Eindrücke



Empfehlungen



Bzgl. der Eindrücke, die den wesentlichen Teil einer Qualitätsbewertung ausmachen, bedeutet dies eine über 91-prozentige Zusage zu den aus dem virtuellen Informationsraum effizient gewonnenen Eindrücken. Die im Verhältnis dazu geringere 66-prozentige Zustimmung zu den Restrukturierungsempfehlungen belegt, daß Restrukturierungsempfehlungen als Kombination von Eindruck und Lösung besonders diskutabel sind: Während die Qualitätsbewertung nach sorgfältiger Spezialisierung des verwendeten Qualitätsmodells noch konsensfähig vornehmbar scheint, ist dies für konkrete, in die Zukunft weisende Restrukturierungen, die zudem häufig von subjektiven und nicht in Qualitätsmodellen ablegbaren Vorlieben bestimmt werden, deutlich schwerer. Allerdings wurden auch die abgelehnten Restrukturierungsempfehlungen als wertvoll angesehen, da sie interessante Alternativen aufzeigten.

Ähnlich positive Ergebnisse wurden in einigen weiteren Projekten dieser Art gewonnen (z.B. eine zweimalige Beurteilung des JWAM-Frameworks). In allen Fällen hat sich gezeigt, daß die Exploration des durch CrocoCosmos erstellten virtuellen Informationsraums ein sehr brauchbares Konzept darstellt, um große Systeme bzgl. der Qualität effizient zu bewerten. Notwendige Voraussetzung war jeweils die möglichst umfangreiche Anpaßbarkeit von CrocoCosmos und eine entsprechende Qualitätsmodellerstellung mit dem Auftraggeber im Vorfeld einer Analyse.

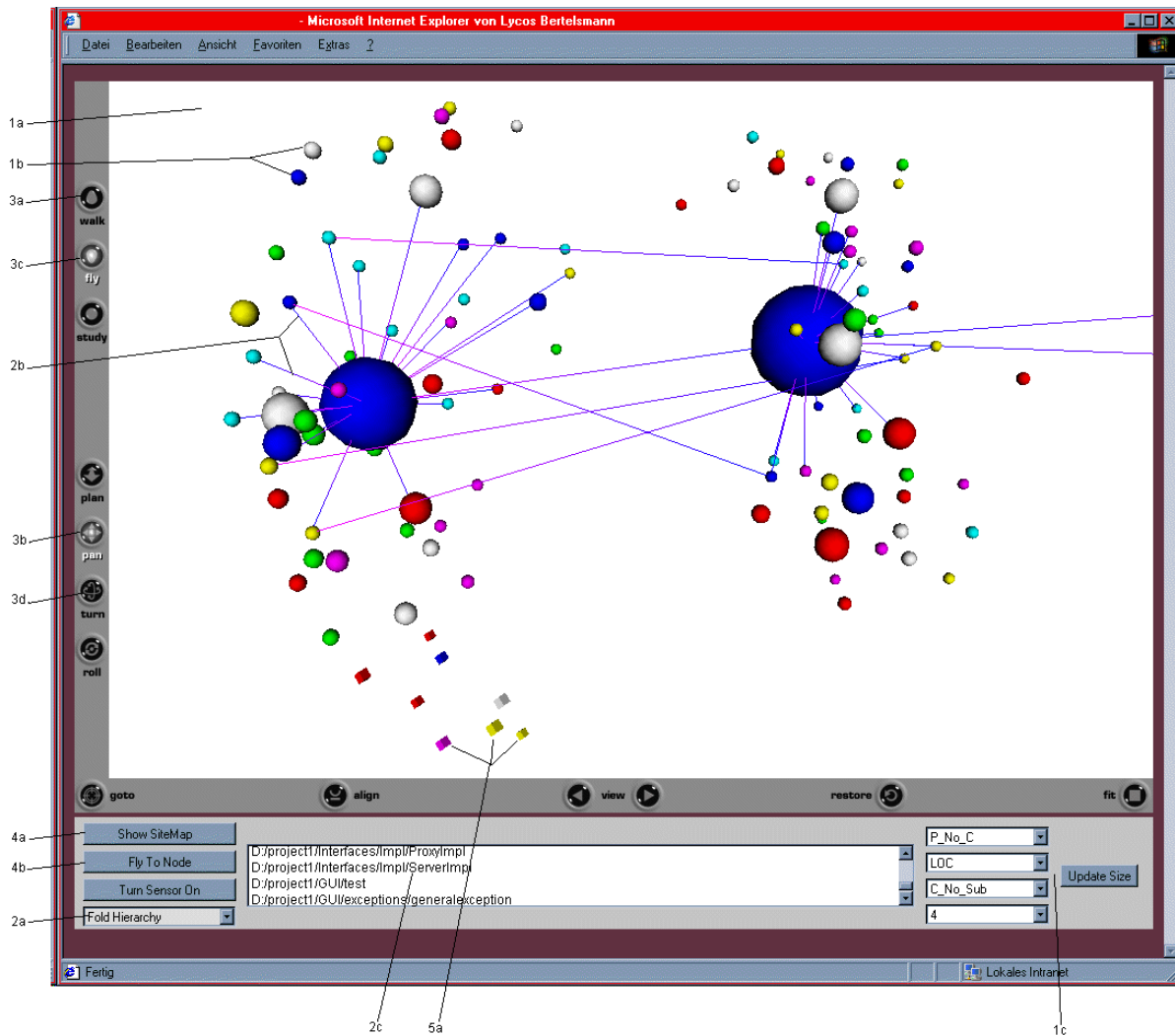
## 5 Zusammenfassung

Virtuelle Informationsräume, die in anderen, große Datenmengen verarbeitenden Bereichen bereits erfolgreich eingesetzt werden, bieten auch für die Qualitätseinschätzung von großen Softwaresystemen eine interessante Möglichkeit, die verschiedenen, relevanten Informationen integriert darzustellen. Der von CrocoCosmos angebotene, anpaßbare Informationsraum hat sich dafür als besonders gut geeignet erwiesen. Die Möglichkeit, mehrere Abstraktionsstufen innerhalb des Systems zu visualisieren, die Möglichkeit der Einblendung konkreter Strukturdaten sowie das anpaßbare Anreichern der Visualisierung mit Meßdaten erlauben eine effiziente Identifikation und Analyse von prägnanten Softwarestrukturen und Systemeigenschaften. Die Balance zwischen globaler Sicht auf das System und detaillierteren Sichten auf selektierbare Systemteile wird dabei durch Techniken wie dynamisch explorierbare Abstraktionsebenen sowie dem der Visualisierung zugrunde liegenden Konzept des generischen Distanzmaßes gewahrt. Letzteres sorgt dafür, daß die relativen Positionen der dargestellten Elemente bereits wertvolle Informationen über spezifische Systemeigenschaften geben. Insbesondere die Anpaßbarkeit des Werkzeugs an von dem Auftraggeber gestellte Qualitätsanforderungen erlaubt ein zielgerichtetes Identifizieren von Systemteilen, die den gestellten Anforderungen nicht genügen. In einer praktischen Anwendung wurde gezeigt, daß CrocoCosmos die effiziente Qualitätsbewertung großer Softwaresysteme ermöglicht, deren Ergebnisse von Systemkennern größtenteils bestätigt werden.

## 6 Referenzen

- [BaWe84] V.R. Basili, D. Weiss: „*A methodology for collecting valid software engineering data*“, IEEE Transactions on Software Engineering, 10(6), Seite 728-738, 1984
- [Chen99] Chaomei Chen: „*Information visualisation and virtual environments*“, Springer-Verlag, London, 1999
- [DäPa98] Rolf Däßler, Hartmut Palm: „*Virtuelle Informationsräume mit VRML: Informationen recherchieren und präsentieren in 3D*“, dpunkt-Verlag, Heidelberg, 1998
- [DeDuLa99] Serge Demeyer, Stephane Ducasse, Michele Lanza: „*A hybrid reverse engineering approach combining metrics and program visualization*“, in Proceedings des Workshops „Experiences in Reengineering“ der European Conference on Object-Oriented Programming (ECOOP) 99, in Lissabon/Portugal, FZI-Karlsruhe, 1999
- [EiStSu99] Stephen G. Eick, Joseph L. Steffen, Eric E. Sumner: „*Seesoft - A Tool for Visualizing Line Oriented Software Statistics*“, in Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman (Hrsg.): „*Readings in Information Visualization - Using Vision to Think*“, Morgan-Kaufmann Publishers, S. 419-430, San Francisco, 1999
- [FaPoFu99] Kim M. Fairchild, Steven E. Poltrock, George W. Furnas: „*SemNet: Three-Dimensional Graphic Representations of Large Knowledge Base*“, in Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman (Hrsg.): „*Readings in Information Visualization - Using Vision to Think*“, Morgan-Kaufmann Publishers, S. 190-206, San Francisco, 1999
- [JaGaRi99] Mehi Jazayeri, Harald Gall, Claudio Riva: „*Visualizing Software Release Histories: The use of color and third dimension*“ in proceedings der International Conference on Software Maintenance (ICSM), 30.8-3.9.99, Oxford, 1999
- [JaLi91] Ivar Jacobson, Fredrik Lindström: „*Re-engineering of old systems to an object-oriented architecture*“, in Andreas Paepcke (Hrsg.): „*Conference on Object-Oriented Programming Systems, Languages, and Applications: 6<sup>th</sup> Annual Conference, Phoenix*“, S. 340-350, Sigplan Notices, Vol. 26, No. 11, November 1991
- [Przy99] Michael Przybilski: „*Integration of metrics visualization into a software development environment*“, Diplomarbeit am Lehrstuhl Software-Systemtechnik der Technischen Universität Cottbus, 1999
- [ScPo98] Heinz W. Schmidt, Margot Postema: „*Reverse Engineering and Abstraction of Legacy Systems*“, in Informatica: „*An International Journal of Computers and Informatics*“, Seite 359-371, Vol. 22, No. 3, 1998
- [SiLöLe99] Frank Simon, Silvio Löffler, Claus Lewerentz: „*Distance based cohesion measuring*“, in proceedings der 2nd European Software Measurement Conference (FESMA) 99, Seite 69-84, Technologisch Instituut Amsterdam, 1999
- [SiRuLe00] Claus Lewerentz, Heinrich Rust, Frank Simon: „*Quality - Metrics - Numbers - Consequences*“, in Reiner Dumke, Franz Lehner (Hrsg.): „*Software-Metriken: Entwicklungen, Werkzeuge und Anwendungsverfahren*“, Seite 51-70, Gabler Verlag, Wiesbaden, 2000
- [StDu] George Stark, Robert C. Durst: „*Using Metrics in Management Decision Making*“, in Paul W. Oman, Shari Lawrence Pfleeger: „*Applying Software Metrics*“, S. 60-66, IEEE Computer Society Press, Los Alamitos, 1997

## 7 Anhang A: Annotierter Screenshot von CrocoCosmos



Dabei bedeuten:

- 1.x Kohäsionsdaten: Dargestellt wird die Objektübersicht (1a) zusammen mit der Objektdarstellung (1b). Die Parameter für die letzteren Daten werden für die jeweilige Abstraktionsstufe (von oben nach unten: Subsystem, Datei, Klasse) unter Verwendung eines Skalierungsfaktors in (1c) eingestellt.
- 2.x Strukturdaten: Mittels des Pull-Down-Menüs (2a) kann zwischen verschiedenen Modi gewählt werden, die jeweils unterschiedliches Verhalten für die Selektion eines Objekts repräsentieren (s.u.). Diese umfassen u.a. die explizite Darstellung von Strukturdaten wie z.B. Vererbungsbeziehungen (2b) und nähere Angaben zur Identifikation (2c).
- 3.x Navigation: Hier können die verschiedenen Navigationstechniken ausgewählt werden.
- 4.x Die zusätzlich mittels (4a) darzustellende Projekthierarchie kann durch Auswahl von (4b) für die Navigation innerhalb der Welt verwendet werden.
- 5.x Abstraktionsniveauselektion: Das Pull-Down-Menü (2a) beinhaltet ebenfalls beide Möglichkeiten der Abstraktionsniveauselektion (global und für einzelne Objekte). Im Screenshot sind unterschiedliche Abstraktionsniveaus durch unterschiedliche Formen symbolisiert (vgl. 5a und 1b). Das Pull-Down-Menü ermöglicht darüber hinaus die Generierung unterschiedlicher Visualisierungen knotenorientierter Maßdaten.