Eine modulare Architektur zur Vereinfachung der Entwicklung klinischer Bildverarbeitungssysteme

Athanasios M. Demiris, Carlos E. Cardenas S., Manuela H. Makabe, Hans-Peter Meinzer

Abt. für Medizinische und Biologische Informatik
Deutsches Krebsforschungszentrum
INF 280, 69120 Heidelberg
Email: A.M.Demiris@dkfz-heidelberg.de

Zusammenfassung. Das Problem der Bearbeitung medizinischen Bildmaterials in der klinischen Routine wurde bisher systemtheoretisch auf zwei Arten angegangen. Die eine Lösung sieht ein monolithisches System vor, das mit einem Maximum an Funktionalität ausgestattet ist. Die andere dagegen besteht aus mehreren kleineren Systemeinheiten, welche spezialisierte Teilaufgaben in der klinischen Bildverarbeitung angehen, wie z.B. Volumenmessung in der Leberchirurgie, Flußmessungen in der Herzchirurgie und zahlreiche andere. Diese kleineren Systeme können dann in ein sog. Host-System eingebettet werden und koexistieren. Für die Unterstützung der Entwicklung der modularen und konfigurierbaren Systeme der zweiten Gruppe stellen wir eine Architektur vor. Wir erläutern die Gründe für die Einführung einer derartigen Architektur, die Aktivitäten des Bildverarbeiters, die diese Architektur unterstützen soll, ihre Komponenten, und abschließend eine mögliche Implementierung in Form von objektorientierten Frameworks.

Schlüsselwörter: objektorientierte Frameworks, Entwicklungswerkeuze, Bildverarbeitungsalgorithmen Repository, Parametrisierung

1 Einleitung

Die Bildverarbeitung hat ihren Weg in viele Anwendungsdomänen gefunden. In einigen dieser Domänen soll ihre Einführung nicht nur eine Erweiterung, sondern auch eine Beschleunigung der Aktivitäten darstellen. Damit ist die zusätzliche Auseinandersetzung der Anwender mit Bildverarbeitungsinternas ausgeschlossen. Das gilt insbesondere für die klinische Bildverarbeitung.

Ein Problem der klinischen Bildverarbeitung ist die Vielfalt der Aufgaben, die damit angesprochen werden können. Dieses wird in vielen Systemen mit einem Maximum an verfügbaren Algorithmen versucht zu lösen. Somit entstehen oft Systeme, welche primär technische und weniger medizinische Kenntnisse erfordern und damit niedrige Akzeptanz erzielen. Eine zweite Lösung ist die komponentenbasierte Entwicklung mehrerer Teillösungen, die anschießend kombiniert werden können.

In unserer Arbeitsgruppe wurde aufgrund der Vielfalt der klinischen Einsatzgebiete die zweite Lösung bevorzugt. Diese Lösung bedeutet das Entwickeln einer dedizierten

Applikation für jede neue klinische Fragestellung. Es ist deshalb wünschenswert, Aufgaben, die eine schnelle Reaktion auf eine neue Anforderung verzögern können, zu vereinfachen. Als zeitaufwendige Aufgaben erweisen sich das Vertraut machen mit neuem Bildmaterial, die Selektion der adäquaten Bildverarbeitungsalgorithmen, die richtige Parametrisierung derselben, die Evaluation ihrer Wirkung und die Erstellung einer graphischen Benutzungsschnittstelle sowohl zum Testen als auch zum Integrieren in die klinische Routine. Zusätzlich sollte diese Architektur alle Komponenten, die von mehreren Systemen benutzt werden können, identifizieren und zur Verfügung stellen, wie z.B. das Laden und Anzeigen der Bilder. Um die Zeit von der Anforderung zum fertigen, klinisch einsetzbaren System zu reduzieren, haben wir eine Architektur entworfen, die diesen Problemen durch Modularität, Erweiterbarkeit und Anpassungsfähigkeit entgegentritt.

2 Methoden und Vorgehen

Das Szenario, welches unterstützt werden soll, ist in Abbildung 1 gezeigt. Der medizinische Endanwender kontaktiert den Systementwickler mit den Anforderungen ①. Dieser sucht nach existierenden Komponenten, die für die Lösung gut geeignet sind ②. Diese Komponenten sind in einem Datenbanksystem gespeichert und werden aus diesem geladen ③. Bei Bedarf werden für selektierte repräsentative Bilder Parametereinstellungen vom System vorgeschlagen ④.

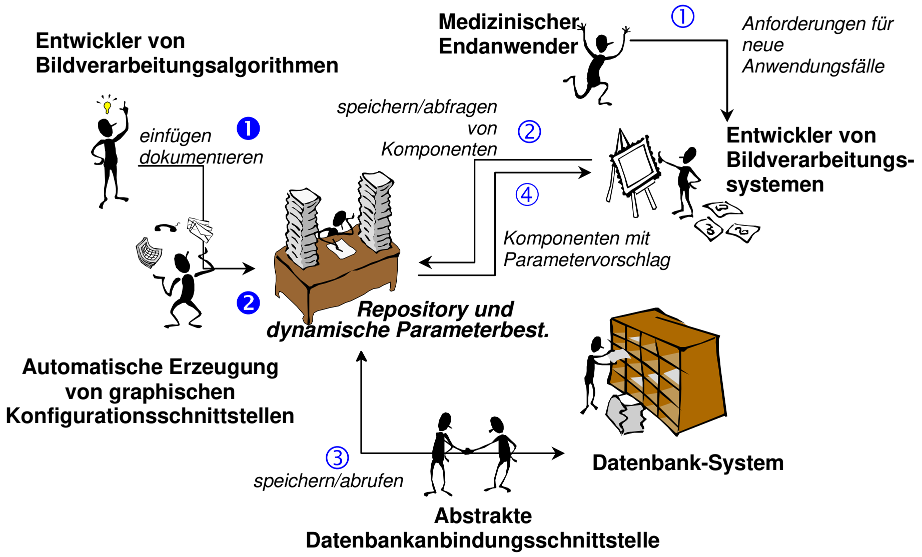

Abb. 1 Übersicht der Abläufe für die Erstellung eines neuen Bildverarbeitungssystems (Schritte ①, ②, ③, ④) und der Erweiterung des Algorithmen-Repository (Schritte ❶, ❷).

Wird ein Algorithmus für ein neues Teilproblem der Bildverarbeitung entwickelt, so kann dieser in das Repository aufgenommen werden. Um seine formale Dokumentation, und damit ein schnelleres Auffinden, zu gewährleisten, werden vom Entwickler zusätzliche Informationen, die im Repository-Modell enthalten sind, eingegeben ➊. Optional kann eine graphische Konfigurationsschnittstelle zum interaktiven Einstellen der Parameter automatisch erzeugt werden und im Repository aufgenommen werden ➋. Dann kann der neue Algorithmus (oder die neue Sequenz von Algorithmen) als eine vollständige, wiederverwendbare Komponente bezeichnet werden.
Im Folgenden werden die Komponenten, die diese Aktivitäten unterstützen sollen, im Einzelnen beschrieben.

2.1 Algorithmen-Repository

Die Wiederverwendung von Software auf der Basis von Programmiersprachen beinhaltet noch den Nachteil, daß die am besten geeigneten wiederzuverwendenden Komponenten nicht leicht zu finden sind. Aus diesem Grund ist eine Strukturierung und das Versehen der Software-Module mit zusätzlicher Information nötig. Die Lösung für dieses Problem ist ein Repository. Existierende Ansätze sind in der Regel entweder zu kompliziert, da sie domänenübergreifend sein sollen [1], oder zu eingeschränkt, weil sie plattformgebunden sind [2].
Aus diesem Grund haben wir ein Repository unter Berücksichtigung existierender Standards, wie CDIF (Computer Aided Software Engineering Data Interchange Format) [3], speziell für die klinische Bildverarbeitung entwickelt. Zunächst wurde eine allgemeine Beschreibung für medizinische Bildverarbeitungsalgorithmen entworfen und implementiert. Diese stellt eine Spezialisierung bekannter Beschreibungsmodelle für allgemeine Algorithmen dar, wie sie z.B. in vielen objektorientierten Programmiersprachen zu finden sind [4], [5], [6] oder in CDIF modelliert wurden. Zusätzlich sind die Strukturen für das Ablegen des notwendigen medizinischen Wissens modelliert. So kann die Selektion der adäquaten Algorithmen auch anhand nicht technischer Kriterien erfolgen.
Das gesamte Modell kann einfach auf eine abstrakte Datenbankschnittstelle abgebildet werden, so daß die am weitesten verbreiteten Datenbanken zum Aufnehmen des Repository angeschlossen werden können.

2.2 Untersuchung des Bildmaterials

Die zweite Komponente behandelt die Untersuchung des Bildmaterials. Nach der Eingabe einer Reihe repräsentativer Bilder kann das System verschiedene Aspekte, wie Grauwerteigenschaften, -profile u.ä, untersuchen und statistisch aufbereiten. Abhängig von den Kenntnissen des Benutzers können diese Ergebnisse in ihrer Gesamtheit oder speziell aufbereitet präsentiert werden. Diese Werte können nun in das Repository aufgenommen werden und als formale Beschreibung der Erfahrung mit einem Algorithmus (oder mit einer Sequenz von Algorithmen) auf einem bestimmten Bildmaterial (z.B. CTAP Bilder der Leber) dienen.

2.3 Vorparametrisierung und Evaluation

Die dritte Komponente übernimmt die optimale Parameter-Einstellung der selektierten Algorithmen in Bezug auf das Bildmaterial. Aus den errechneten Eigenschaften der Bilder kann ein initialer Parametersatz für einen Algorithmus bestimmt und anschließend dessen Wirkung mit diesen Parametern evaluiert werden.
Die Evaluation erfolgt mit Hilfe einer Fuzzy-Logik-basierten Steuerung und kann allerdings nur dann eingesetzt werden, wenn statistische Werte über die Kategorie vom aktuell vorliegenden Bildmaterial vorhanden sind, die als Richtwerte dienen. Wenn die Evaluation schlecht ausfällt, werden die Werte angepaßt und ein neuer Versuch erfolgt.
Zur Evaluation von Segmentierungsalgorithmen stehen momentan folgende Möglichkeiten zur Verfügung: relative Größe des segmentierten Objekts zu einem einfach zu detektierenden Objekt in der selben anatomischen Umgebung, Schwerpunktlage, Ausbreitung des Objekts, Richtung des Trägheittensors, Zusammenhang (grauwertbasiert), Homogenität.

2.4 Automatische Erzeugung von graphischen Konfigurationsschnittstellen

Die vierte Komponente stellt aufgrund der Algorithmusschnittstelle (Signatur) automatisch graphische Benutzungselemente zur Verfügung, um die Algorithmen zu testen und anschließend in eine übergeordnete Applikation zu integrieren. Die Parametertypen werden herangezogen, um für jeden Parameter ein dediziertes, adäquates Element zu erzeugen. Die Algorithmen können momentan in C oder C++ implementiert sein und die Datentypen des in unserer Abteilung verwendeten Bildformats einsetzen. Sowohl die Erweiterung auf andere Implementierungssprachen der Algorithmen als auch zusätzlich definierte Datentypen lassen sich durch leichte Modifikationen der Komponente erreichen. Um eine größere Flexibilität zu gewährleisten, basierte der Entwurf dieser Komponente auf dem MVC-Konzept, welches die eigentliche Applikation von der graphischen Benutzungsschnittstelle trennt [7].
Das Aussehen der graphischen Elemente kann von der Host-Applikation festgelegt werden, indem diese einen Satz von solchen, an sie angepassten Elementen zur Verfügung stellt. Alternativ ist es möglich, ein von der Komponente angebotenes Aussehen zu wählen. Die Gestaltung der Komponenten erfolgte nach Regeln der kognitiven Ergonomie in der medizinischen Informations- und Bildverarbeitung [8], [9].

2.5 Eine mögliche Implementierung mit objektorientierten Frameworks

Die Beschreibung der einzelnen Komponenten der hier vorgestellten Architektur soll flexibel gehalten werden und konfigurierbar sein. Da objektorientierte Frameworks diese Bedingungen optimal erfüllen, scheint ihre Verwendung am besten geeignet zu sein [10], [11]. Aus diesem Grund haben wir für jede Komponente jeweils ein Framework implementiert. Somit lassen sich alle Einzelteile getrennt voneinander modifizieren und an andere Umgebungen anpassen. In ihrer Gesamtheit bleiben sie trotzdem funktionstüchtig.

3 Diskussion und Ausblick

Die Komponenten der hier beschriebenen Architektur sind in unterschiedlichen Ausbaustufen implementiert und die erste Machbarkeitsstudie ist erfolgreich abgeschlossen. Das Repository ist bereits modelliert und in der Programmiersprache Smalltalk implementiert. Eine Portierung nach Java ist bereits angelaufen. Eine Reihe von Algorithmen zur Vorparametrisierung und Evaluation wird an Einzelfällen getestet. Bislang haben wir uns auf die Bildgebungsmodalitäten MR und CT und die anatomischen Regionen Abdomen und Herz konzentriert. Die automatische Erzeugung von graphischen Konfigurationsschnittstellen ist in ihrer ersten Phase abgeschlossen. Eine Ansammlung von statistischen Auswertungsalgorithmen wird mit dem Repository gekoppelt, um sie über Begriffe der Bildverarbeitung anzusteuern.
Da eines der Ziele die Erweiterbarkeit war, haben wir die Komponenten als objektorientierte Frameworks umgesetzt. Noch fehlende Elemente können deswegen einfach hinzugefügt werden. Um eine Verteilung zu gewährleisten, werden einige der Komponenten mit CORBA-Funktionalität ausgestattet [12]. Eine größere Studie, die den eigentlichen Zeitgewinn nach Einsatz der hier beschriebenen Architektur untersucht, ist geplant. Der routinemässige Einsatz sollte in den nächsten Monaten stattfinden.

4 Literatur

1. Unisys Corp. Universal Repository Information Model Technical Overview. Doc. Nr. 86002094, 1996.
2. Bernstein PA, Sanders P, Harry B, Shutt D, Zander J: The Microsoft Repository. Proceedings of the 23rd VLDB Conference, Athens Greece, 1997.
3. EIA/IS-106: CDIF CASE Data Interchange Format - Overview. Released as Interim Standard.
4. Goldberg A, Robson D: Smalltalk-80 The Interactive Programming Environment. Addison-Wesley, Reading MA, 2. Auflage 1989.
5. Grand M, Knudsen J: Java Fundamental Classes Reference. O'Reilly & Assoc., Sebastopol CA, 1997.
6. Cattell RGG (Ed): The Object Database Standard: ODMG-93 Release 1.1. Morgan Kaufmann Publishers, 1994.
7. Rumbaugh J: Modeling Models and Viewing Views: A Look at the Model-View-Controller Framework. Journal of Object-Oriented Programming, 7(2):14-20, 1994.
8. Borälv E, Göransson B, Olsson E, Sandblad B: Usability and Efficiency: the HELIOS Approach to Development of User Interfaces. Computer Methods and Programs in Biomedicine, 45(Suppl.):47-64, 1994.
9. Demiris AM, Meinzer HP: Cognition Based Development and Evaluation of Ergonomic User Interfaces for Medical Image Processing and Archiving Systems. Medical Informatics, in print.
10. Lewis T (Ed): Object-oriented application frameworks. Manning Publications, Greenwich, 1995.
11. Taligent Inc.: Leveraging Object-Oriented Frameworks. Taligent Inc., Cupertino CA, 1993.
12. Object Management Group, Inc.: Common Object Request Broker Architecture and Specification, CORBA Revision 2. Framingham, Massachusetts, 1995.