

Ein generisches Interaktionskonzept mit Undo für die medizinische Bildverarbeitung

Ingmar Wegner, Marcus Vetter, Ivo Wolf und Hans-Peter Meinzer

Abteilung Medizinische und Biologische Informatik
Deutsches Krebsforschungszentrum, D-69120 Heidelberg
Email: i.wegner@dkfz.de

Zusammenfassung. Mit Hilfe der rechnergestützten Bildverarbeitung können Ärzte besser analysieren und diagnostizieren, aber auch Operationen präziser planen und durchführen. Dabei wird häufig eine komplexe Mensch-Maschine-Interaktion benötigt, um das gewünschte Resultat zu erzielen. Fehler bei der Bedienung können nicht ausgeschlossen werden und so ist ein geschicktes Zusammenspiel zwischen Interaktion und Rücknahmemöglichkeit eines Befehls hilfreich. Dieser Beitrag beschreibt, wie mittels Zustandsmaschinen ein allgemeines, umfangreiches und generisches Interaktionsmuster aufgebaut und über XML konfiguriert werden kann. Ein allgemeiner Undo-Mechanismus, der optional genutzt werden kann, ist in das Interaktionskonzept integriert.

1 Einleitung

Verfahren der medizinischen Bildverarbeitung sehen häufig eine komplexe Wechselbeziehung zwischen Mensch und Computer vor. Diese Mensch-Maschine-Interaktion ermöglicht dem Benutzer beispielsweise ein geometrisches Objekt in einer dreidimensionalen Ansicht zu verschieben. Dabei kann dies durch eine unterschiedliche Abfolge von Befehlen geschehen. Diese Abfolge kann sich von Benutzer zu Benutzer unterscheiden, so dass es von Vorteil ist, wenn die Interaktion auf den Benutzer anpassungsfähig ist. Zum Beispiel kann ein Benutzer das direkte Manipulieren eines Objekts ohne vorheriges Selektieren präferieren. Einige Softwareprodukte wie z.B. SurgeryPlanner von MeVis (www.mevis.de) bieten diese benutzerspezifische Definition der Interaktion an. Jedoch wenn nicht nur eine sondern verschiedene Applikationen mit unterschiedlichen Anforderungen bei Wiederverwendung der Interaktionsmechanismen erstellt werden sollen, wird ein allgemeines und umfassendes Konzept der Interaktion notwendig. Dabei bietet sich eine generische Realisation der Interaktion an, in der erst zu Beginn der Applikation die genaue Abfolge der Interaktion festgelegt wird. Ein schnelles Anpassen der Interaktion sowie das Laden von Benutzerpräferenzen wird hierdurch erleichtert.

Während der Interaktion kommt es häufig vor, dass der Benutzer einen Fehler begeht [1]. Benutzerfreundlich ist ein Programm, wenn es dem Benutzer ermöglicht, diesen Fehler rückgängig zu machen. Diese Möglichkeit der Rücknahme und Wiederholung eines Befehls (Undo/Redo) hat sich in fast allen interaktiven Systemen (Textverarbeitung, Grafikdesign etc.) durchgesetzt, jedoch

bisher kaum in der medizinischen Bildverarbeitung. Dabei fördert sie die Akzeptanz der überwiegend komplexen Anwendungen. Die Erweiterung einer Applikation mit Undo/Redo-Funktionalität ist nicht trivial und deswegen muss die Funktionalität bereits in der Planung berücksichtigt werden.

Es existieren mehrere Ansätze, wie die nötigen Informationen für das Undo und das Redo eines Befehls aufbewahrt werden. Das eingeschränkte lineare Undo-Modell z.B. legt die Information in einem Kellerspeicher (Undo-Stack) ab, so dass alle Befehle der Reihe nach gespeichert werden. Der Benutzer kann nach Belieben die Befehle in der umgekehrten Reihenfolge rückgängig machen, woraufhin dieser Befehl in einem Redo-Speicher abgelegt wird. Bei Neueingabe eines Befehls wird der Redo-Speicher gelöscht. Ein weiteres Modell, das Single-Undo-Modell speichert nur die Information des zuletzt getätigten Befehls ab.

Die Information, wie ein bestimmter Befehl rückgängig gemacht wird, kann in einer inversen Operation abgelegt werden [2]. Zu einer Operation „Punkt hinzufügen“ wäre die Inverse „Punkt löschen“. Weitere Ansätze sehen ein erneutes Ausführen der Applikation bis zur erwünschten Befehlsposition vor.

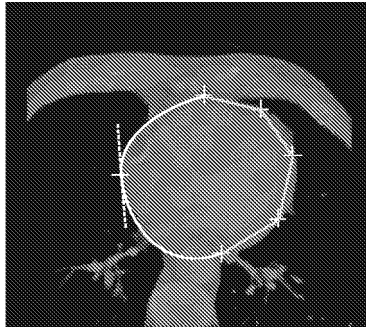
Trolltech stellte im Dezember 2003 ein Undo/Redo-Framework begleitend zu dem GUI-Toolkit QT vor, welches sich an den Command-Design-Pattern von QT orientiert. Das Undo-Modell richtet sich nach dem eingeschränkten limitierten Undo-Modell. Ein allgemein formulierter Ansatz eines Open-Source-Toolkits, welcher den Einsatz unterschiedlicher Undo-Modelle vorsieht, ist bisher nicht bekannt.

Das hier vorgestellte Konzept kombiniert eine umfangreiche und anpassungsfähige Interaktion mit einem erweiterbaren Undo-Mechanismus und wurde in das Open-Source-Projekt Medical Imaging Toolkit (MITK) [3] integriert. Das im Deutschen Krebsforschungszentrum Heidelberg entwickelte MITK unterstützt die Erstellung von interaktiven medizinischen Bildverarbeitungsprogrammen mit Bedarf an komplexer Interaktion. Es vereint die Segmentierungs- und Registrierungs-Algorithmen vom Insight Toolkit (ITK) mit den Visualisierungs-Algorithmen vom Visualisation Toolkit (VTK) und fügt u.a. Funktionalität zur Datenverwaltung hinzu.

2 Methoden

Das Problem der Realisation einer komplexen Interaktion wird durch die Zerlegung in Teilinteraktionen, die hierarchisch angeordnet werden können, gelöst. Beispielsweise kann die Interaktion eines geometrischen Objekts in Subfiguren, Linien und Kurven, zerlegt werden (s. Abb. 1). Die Modellierung der Interaktion der Subfiguren gestaltet sich wesentlich leichter, da zum einen weniger Verhalten abgebildet werden muss und zum anderen feste Grenzen zwischen den Abstraktionsebenen existieren. Ein Kurvenverlauf besteht aus einzelnen Kurven, die wiederum aus Stützstellen bestehen. Die Interaktion der Stützstellen teilt sich in das Verhalten von Tangenten und Punkten auf. Die Punkte sind in dieser Hierarchie das unterste System und können gesetzt, gelöscht, verschoben und selektiert werden.

Abb. 1. Beispiel eines Objekts bestehend aus einem Bezierkurvenverlauf(links) und einem Streckenverlauf (rechts). Punkte haben gleiches Verhalten, können aber z.B. um das Verhalten einer Tangente (links) erweitert werden. Das aus unterschiedlichen Subobjekten zusammengesetzte Objekt kann für die Segmentierung herangezogen werden.



Durch die Formulierung der Interaktion in mehrere Abstraktionsebenen können diese wiederverwendet werden. Ein Linienverlauf benötigt die Interaktion von Punkten genauso wie ein Kurvenverlauf. Weiterhin können die Interaktionen durch ihre Zerlegung parallel und unabhängig voneinander entwickelt werden.

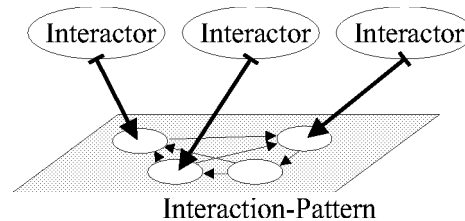
Die Interaktion wird objektorientiert durch Mealy-Zustandsmaschinen [4] implementiert. Diese bestehen aus Zuständen, Übergängen auf einen nächsten Zustand, Ereignissen und auszuführenden Aktionen. Eine deterministische Zustandsmaschine befindet sich zu jeder Zeit in genau einem Zustand, hier realisiert über einen Zeiger, und wartet auf ein Ereignis, das vom Benutzer beispielsweise durch einen Tastendruck ausgelöst wird. Empfängt die Maschine ein Ereignis, wird im aktuellen Zustand nach dem entsprechendem Übergang gesucht und bei Erfolg der Zeiger auf den nächsten Zustand gesetzt. Dabei wird eine im Übergang definierte Aktion, die beispielsweise einen Punkt verschiebt, ausgeführt. Die Maschine befindet sich nun in einem neuen Zustand, der eine neue Anordnung von Übergängen mit dazugehörigen Ereignissen beinhaltet. Durch dieses Konzept kann das komplexe Verhalten beliebiger Interaktion optimal abgebildet werden.

Alle Definitionen der Zustandsmaschinen werden in schriftlicher Form in einer XML-Datei abgelegt. Dabei können auch unterschiedliche Konfigurationen abgelegt werden, was das Laden von Benutzerpräferenzen ermöglicht.

Die Realisation der Interaktion wird durch zwei Untereinheiten bewerkstelligt: In dem Interaktionsobjekt wird eine Referenz auf den aktuellen Zustand gehalten und es werden alle Aktionen implementiert. Das Verhaltensmuster bildet ein Konstrukt aus Objekten, das die Zustände und Übergänge mit den dazugehörigen Informationen über Ereignis und Aktion darstellt.

Zum Programmbeginn werden alle Zustandsmuster durch die Definitionen in der Datei erzeugt. Während des Programmablaufs können dann mehrere Interaktionsobjekte unterschiedliche Zustände eines Zustandsmusters referenzieren (Abb. 2), wodurch wertvolle Ressourcen gespart werden können. Durch den generischen Ansatz verlangt eine Änderung des Verhaltens keine Neuerstellung des Programms, sondern lediglich einen Neustart.

Abb. 2. Wiederverwendung eines Interaktionsmusters. Zugriff von mehreren Interaktionsobjekten (Interactor) auf ein Zustandsmuster (Interaction-Pattern).



Die Erstellung einer neuen Interaktion kann auf drei verschiedene Weisen geschehen:

- Veränderung eines bestimmten Ereignisses eines Übergangs (XML-Datei)
- Anpassung der Abfolge in einer Zustandsmaschine (XML-Datei)
- Ableitung einer Klasse und Ergänzung der fehlenden Aktion

Eng in das Interaktionskonzept integriert, aber trotzdem optional gehalten, zeigt sich das Undo-Konzept. Es ist gekapselt, wodurch unterschiedliche Undo-Modelle unterstützt werden. Im Einsatz steht bisher das weit verbreitete eingeschränkte lineare Undo-Modell.

Jede Veränderung der Daten wird über ein Operationsobjekt bewerkstelligt. In diesem Objekt werden alle für die Veränderung sowie für die Rücknahme der Veränderung nötigen Informationen gespeichert. Dies schließt ebenso die Identifikationsnummer des verursachten Benutzerbefehls mit ein, so dass alle zugehörigen Operationsobjekte detektiert werden können.

Die Objekte werden von einer Undo-Kontrolleinheit verwaltet und abhängig vom gewählten Undo-Modell in Undo- bzw. Redo-Liste abgelegt. Zur regulären Ausführung der Veränderung werden nur die für die Veränderung notwendigen Informationen an das entsprechende Datum übergeben, welches dann die Veränderung durchführt.

Nach Belieben kann der Benutzer einen Undo-Befehl erteilen, wonach alle für die Rücknahme der Veränderung nötigen Informationen an das entsprechende Datum gesendet werden. Dies wird für alle in der Undo-Liste befindlichen Operationsobjekte eines Benutzerbefehls durchgeführt. Hierauf werden die Objekte in die Redo-Liste übertragen und für folgende Redo-Befehle bereitgehalten.

Zuzüglich der Befehls-ID wird eine Gruppen-ID in den Operationsobjekten abgelegt. Hierdurch ist das Rückführen mehrerer Befehle möglich, so dass nicht nur das Setzen eines Punktes, sondern das gesamte Aufbauen einer Punktwolke rückgängig gemacht werden kann. Hierdurch kann schneller zum gewünschten Arbeitsschritt zurückgegangen werden.

3 Ergebnisse

Das Interaktionskonzept wurde für vielseitige Anwendungen der medizinischen Bildverarbeitung entwickelt und erlaubt eine konsistente Interaktion in 2D sowie

3D. Derzeit wird es im MITK erfolgreich eingesetzt und zeigt bereits die erwarteten Vorteile der Wiederverwendung von Interaktionskomponenten. Zum einen bleibt dem Entwickler eine Doppelimplementierungen erspart und zum anderen kann der Benutzer bei neuen Interaktionen auf bereits bestehendes Wissen zurückgreifen.

Die schnelle Anpassung der Interaktion durch Veränderung der XML-Datei hat sich als vorteilhaft für die Erprobung neuer Interaktion erwiesen. Gerade bei großen Software-Projekten lässt sich ein klarer Zeitvorteil feststellen.

Durch die Formulierung einer inversen Operation stellt sich das Undo-Konzept für den Entwickler als eigenständig dar. Die Funktionalität steigert die Akzeptanz des Benutzers und ermöglicht ein bequemes Arbeiten, bei dem ein Fehler nicht zum Neubeginn der Arbeit führt.

4 Diskussion

Die Definition der Zustandsmaschine muss bisher in schriftlicher Form modelliert werden. Entwicklungen an Zustandsmaschinen können jedoch besser in grafischer Form veranschaulicht werden. Das Niederschreiben einer gezeichneten Maschine ist für den Entwickler unnötige Mehrarbeit, die in Zukunft durch einen grafischen Editor abgenommen werden soll.

Ist eine Formulierung einer inversen Operation für ein Undo nicht möglich, so liegt die Entscheidung beim Entwickler einen Zwischenstatus abzuspeichern oder das Datum bei einem Undo neu zu berechnen. Das zusätzliche Belegen einer Speicherplatz-Ressource ermöglicht dem Benutzer das Rückführen aller Befehle und ist somit vertretbar.

Literaturverzeichnis

1. Brown PS, Gould JD: An Experimental Study of People Creating Spreadsheets. ACM Transactions on Office information Systems 5(3): 258-272, 1987
2. Archer JE, Conway R, Schneider FB: User recovery and reversal in interactive systems. ACM Transactions on Programming languages and Systems 6(1):1-19, 1984
3. Wolf I, Vetter M, Wegner I et. al.: The Medical Imaging Interaction Toolkit (MITK) in Print SPIE 2004 Nr. 194
4. Mealy GH: A method for synthesizing sequential circuits. Bell System Technical Journal 34: 1045-1079, 1955