

17 IDA – Interactive Digital Atlas

Kap. 17 beschreibt die Konzeption und Umsetzung des Interactive Digital Atlas IDA. Die Darstellung gliedert sich in Systemidee und Anforderungen (Kap. 17.1), Grobkonzept (Kap. 17.2), Feinkonzept (Kap. 17.3), den Entwurf der Programmlogik (Kap. 17.4) sowie eine Übersicht über die Interaktionsmöglichkeiten (Kap. 17.5).

17.1 Systemidee und Anforderungen

Zur Unterstützung der Anwender durch eine intuitive Bereitstellung von Raumbezügen wurde anhand entsprechender Vorgaben der Scientific Data Management Group und in enger Zusammenarbeit mit dieser der Interactive Digital Atlas (IDA) konzipiert und realisiert. Es sollte eine netzwerkfähige, graphisch-interaktive Komponente entwickelt werden, die sowohl eine Navigation über verschiedenen Hierarchien geographischer Karten wie eine flexible Selektion von Raumbezügen aus diesen ermöglicht. Die Komponente sollte zudem auf die interaktive Visualisierung georeferenzierter Daten erweiterbar sein. Der Entwicklung von IDA wurden die nachfolgend beschriebenen Anforderungen zugrunde gelegt.

▪ Interaktion mit Karten aus unterschiedlichen Hierarchien

IDA soll dem Anwender die Interaktion mit graphischen Darstellungen von geographischen Karten ermöglichen. Dabei sind unterschiedliche, jeweils hierarchisch strukturierte Sammlungen von Karten (Kartenhierarchien) zugänglich zu machen. Die zu realisierenden Interaktionsformen umfassen Navigation, Zooming sowie Selektion und sollen durch geeignetes Echtzeit-Feedback unterstützt werden.

- | | |
|------------|--|
| Navigation | ▶ Unter <i>Navigation</i> wird hier das nutzerdefinierte Wechseln zwischen einzelnen Karten verstanden. Der Anwender soll zu beliebigen Karten jeder Hierarchie navigieren können. Innerhalb jeder Kartenhierarchie soll die Navigation zu jeweils verfügbaren Karten der nächsthöheren und nächsttieferen Hierarchieebene unmittelbar durch Interaktion mit der graphischen Darstellung einer Karte möglich sein. |
| Zooming | ▶ Unter <i>Zooming</i> wird hier die interaktive Bereitstellung der vergrößerten Darstellung eines Kartenausschnittes verstanden. Zooming soll für nutzerdefinierbare Ausschnitte einer beliebigen Karte möglich sein und direkt durch Interaktion mit ihrer graphischen Darstellung erfolgen können. |
| Selektion | ▶ Unter <i>Selektion</i> wird hier die nutzerdefinierbare Auswahl eines Raumbezuges verstanden. Selektion soll ebenfalls durch Interaktion mit der graphischen Darstellung einer Karte möglich sein und sowohl die direkte Auswahl vordefinierter Einheiten wie die Definition frei definierbarer rechteckiger Raumbezüge (Boundingboxen) umfassen. |
| Feedback | ▶ Der Nutzer soll auf beliebige Punkte der graphischen Darstellung einer Karte zeigen können, um unmittelbar Informationen über geographische Einheiten sowie die dieser Position entsprechenden geographischen Koordinaten abrufen zu können. Dieses <i>Feedback</i> soll in Echtzeit, also ohne wahrnehmbare Verzögerung, bereitgestellt werden. |

▪ Netzwerkfähigkeit und Portabilität

Um eine breite Verfügbarkeit zu ermöglichen, soll IDA - ebenso wie die anderen Bestand-

teile der Schnittstelle - sowohl ortsunabhängig einsetzbar (netzwerkfähig) sowie ablauffähig auf unterschiedlichen Hardware- und Betriebssystemplattformen (portabel) sein.

▪ **Adaptivität**

IDA soll unterschiedliche Kartenhierarchien zugänglich machen können und entsprechend konfigurierbar sein. Die Einbindung neuer sowie die Veränderung oder Erweiterung bestehender Kartenhierarchien soll dabei ohne Reprogrammierung erfolgen können.

▪ **Performance**

Die Performance einer Anwendung ist generell ein wesentliches Kriterium für die Nutzerakzeptanz und von besonderer Relevanz für den intuitiven Umgang mit graphisch-interaktiven Applikationen. Entsprechend ist eine durchgängig hohe Performance von IDA sowohl bezüglich Ladezeiten wie bei der Umsetzung aller Interaktionen sicherzustellen.

▪ **Datengrundlage**

Die Bereitstellung der von IDA zu verarbeitenden Karten soll basierend auf vorhandenen Daten aus Geoinformationssystemen erfolgen. Für eine effiziente Bereitstellung der Karten wurde in Zusammenarbeit mit der Scientific Data Management Group ein geeignetes Datenformat entwickelt. Die für die Datenbereitstellung erforderlichen Prozesse von Aufbereitung und Transformation der Ausgangsdaten in dieses Format sind dabei *nicht* Aufgabe der Komponente.

17.2 Grobkonzept

17.2.1 Webfähiger Java-Client

IDA wurde als auf dem Rechner des Anwenders ablauffähige Client-Anwendung konzipiert. Die erforderliche Datengrundlage wird zentral bereitgestellt und kann von IDA zur Laufzeit über das Internet dynamisch geladen werden²⁶⁸. Die Entwicklung erfolgte unter Verwendung der Programmiersprache Java, um einen plattformunabhängigen Einsatz mit hohem clientseitigen Interaktionsumfang realisieren zu können. Die Komponente wurde dabei so entworfen, dass eine möglichst flexible Anwendbarkeit möglich ist. So sollte sie sowohl unaufwendig in Filtermodule wie Auswertungsmodule der Schnittstelle einbettbar sein; ebenso sollte sie als autonome Anwendung (als Applikation ebenso wie als Applet) sowie in Kombination mit HTML-Formularen oder weiteren Applikationen betrieben werden können (vgl. Abb. 17.1).

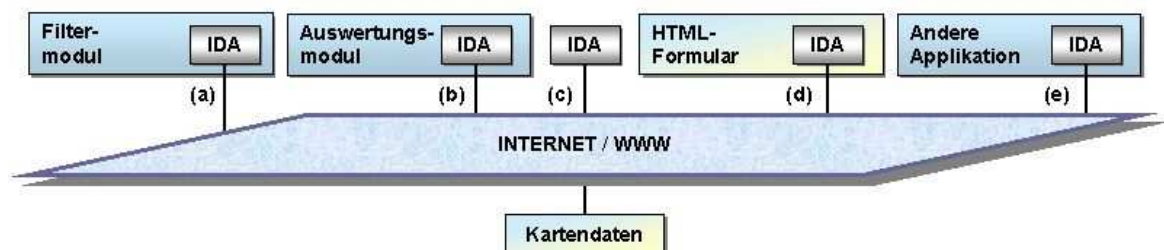


Abb. 17.1 - Beispiele für unterschiedliche Möglichkeiten zum Einsatz der Komponente IDA: (a) und (b) als Teil eines Filtermoduls bzw. eines Auswertungsmoduls der Schnittstelle; (c) als autonome Anwendung; (d) in Kombination mit einem HTML-Formular; (e) als Teil anderer Applikationen.

17.2.2 Zweidimensionale Kartendarstellung

Für die Visualisierung der einzelnen Karten standen zwei- oder dreidimensionale Darstellungsformen zur Diskussion. Die Entscheidung fiel aus folgenden Gründen für eine zwei-

²⁶⁸ Alternativ kann eine lokale Ablage der Kartendaten auf dem Rechner des Anwenders erfolgen (vgl. Kap. 17.3.2).

mensionale Kartenvisualisierung:

1. Zweidimensionale Karten sind den meisten Anwendern aus Atlanten und gegebenenfalls auch aus Geoinformationssystemen vertraut und damit schnell interpretierbar. Sämtliche für die intuitive Orientierung über geographische Bezüge und die geforderten Interaktionen notwendigen Informationen können auf diese Weise anschaulich dargestellt werden.
2. Eine dreidimensionale Darstellung etwa von Oberflächenreliefs ist für die gegebenen Anforderungen nicht erforderlich und birgt zudem Gefahren einer unnötigen Informationsüberflutung und Desorientierung des Anwenders. Dreidimensionale Visualisierungen konfrontieren den Anwender überdies mit den bekannten Problemen von Perspektive und Verdeckung (vgl. Kap. 4.3.2) und können so Orientierung und Interaktion erschweren.

17.2.3 Verwendung von Vektordaten

Der geforderte Funktionsumfang setzt voraus, dass die Komponente Zugriff auf Informationen über die *Umriss*e der zu verarbeitenden geographischen Einheiten²⁶⁹ besitzt. Aus diesem Grund wurden *Vektordaten* als Grundlage der einzelnen Karten gewählt. Vektordaten erlauben die Beschreibung und Identifikation einzelner Polygone und bieten so die Basis, um Koordinaten in Bezug zu diesen zu setzen und erforderliche Innen-Außen-Relationen zu berechnen. Auf Vektordaten basierende graphische Darstellungen (Vektorgrafiken) bieten zudem den Vorteil der Skalierbarkeit, so dass ihre Vergrößerung möglich ist, ohne dass die von Pixelgrafiken bekannten Treppeneffekte auftreten.

17.2.4 Clientseitige Interaktionsverarbeitung und Bildgenerierung

Um die gegebenen Anforderungen an Echtzeit-Feedback sowie performante Interaktion mit graphischen Darstellungen umsetzen zu können, wurde entschieden, die erforderlichen Vektordaten clientseitig, also auf dem Rechner des jeweiligen Anwenders, zu verarbeiten. Die clientseitige Verarbeitung umfasst dabei neben der Umsetzung von Interaktionen und der Bereitstellung von Feedback auch die eigentliche Bildgenerierung aus den Vektordaten. Auf diese Weise wird ein doppelter Transport von Vektordaten und entsprechenden Bildern zum Client überflüssig; zudem ist so eine Flexibilisierung der graphischen Darstellung einer Karte ohne zusätzliche Netzkontakte möglich.

17.3 Feinkonzept

17.3.1 Herausforderungen

Anforderungen und Grobkonzept führten zu zwei zentralen Zielen. Das von der Komponente auf den Rechner des Anwenders zu ladende und dort zu verarbeitende Datenmaterial muss zunächst die Generierung hinreichend detaillierter graphischer Darstellungen von Karten erlauben und zudem klein genug sein, um seinen performanten Transport über das Netz zu ermöglichen. Zugleich muss es eine generische Adaption der Komponente an die ihr zur Verfügung gestellten Karten und die Struktur der jeweiligen Kartenhierarchien unterstützen. Um beide Ziele zu erreichen, wurden die nachfolgend beschriebenen Lösungsstrategien eingesetzt.

17.3.2 Sukzessives Laden und clientseitiges Caching

Um eine flexible Bereitstellung des erforderlichen Datenmaterials anhand der jeweiligen Anforderungen des Anwenders zu ermöglichen, werden die Daten jeder Karte separat vor-

²⁶⁹ Zur verwendeten Begrifflichkeit vgl. Kap. 17.3.3

gehalten. Während die Komponente zum Rechner des Anwenders geladen wird, verbleibt die überwiegende Mehrheit der den Karten zugrundeliegenden Daten zunächst beim Server; Daten werden erst dann über das Netzwerk zum Client geladen, wenn sie für die Bereitstellung einer vom Nutzer angeforderten Karte erforderlich sind. Alternativ dazu besteht die Möglichkeit, die Kartendaten vollständig auf dem Rechner des Anwenders abzulegen, um so diesbezügliche Netzwerkzugriffe überflüssig zu machen. Die Daten einer einmal zur Laufzeit geladenen Karte werden von der Komponente in einem Programm-Cache vorgehalten, so dass ein erneuter Zugriff ohne nochmaliges externes Laden erfolgen kann.

17.3.3 Selbstbeschreibende Karten

Die geforderte Unabhängigkeit der Komponente von konkreten Karten und Kartenhierarchien wurde durch die Entwicklung eines geeigneten Datenformats durch die Scientific Data Management Group erreicht. Das Datenformat stellt die erforderlichen Daten jeder Karte in einer Form bereit, die eine automatische Interpretation erlaubt und ermöglicht es, jede Karte unabhängig von ihren Details und ihrer Position in einer Hierarchie gleichartig zu behandeln. An dieser Stelle sollen zunächst einige Begriffsklärungen vorgenommen werden:

- **Kartenhierarchien**

Eine hierarchisch strukturierte Sammlung miteinander in Zusammenhang stehender Karten wird hier als *Kartenhierarchie* bezeichnet. Dabei kann jede Karte eindeutig einer Ebene der Hierarchie zugeordnet werden. Eine administrative Hierarchie besteht bspw. aus einer Weltkarte, Karten der Kontinente, Länderkarten etc.

- **Geographische Einheiten**

Die vordefinierten Entitäten, die auf einer Karte für den Anwender selektierbar sein sollen, werden hier als *geographische Einheiten* bezeichnet. Jede Karte besteht aus mindestens einer, in der Regel jedoch mehreren geographischen Einheiten. So sind bspw. die geographischen Einheiten einer administrativen Weltkarte die Kontinente, die geographischen Einheiten einer administrativen Deutschlandkarte die einzelnen Bundesländer.

- **Polygone**

Die Vektordaten, anhand derer die Umriss der geographischen Einheiten einer Karte beschrieben werden, sind in *Polygonen* organisiert. Eine geographische Einheit besteht aus mindestens einem Polygon, kann jedoch auch mehrere Polygone umfassen. So gehören bspw. auf einer administrativen Deutschlandkarte zur geographischen Einheit Mecklenburg-Vorpommern mehrere Polygone, die jeweils die Umriss des Festlandes sowie einer Reihe zugehöriger Inseln wie Rügen und Hiddensee beschreiben.

- **Übergeordnete Karten**

Diejenige Karte, die einer individuellen Karte in der nächsthöheren Hierarchieebene übergeordnet ist, wird als deren *übergeordnete Karte* bezeichnet. Jeder Karte, die sich nicht auf der obersten Hierarchieebene befindet, ist genau eine übergeordnete Karte zugeordnet; mehrere Karten können dabei dieselbe übergeordnete Karte besitzen. In einer administrativen Hierarchie stellt beispielsweise eine Europakarte die übergeordnete Karte sowohl für eine Deutschlandkarte wie für eine Frankreichkarte dar; die der Europakarte übergeordnete Karte ist eine Weltkarte, die ihrerseits *keine* übergeordnete Karte besitzt.

- **Untergeordnete Karten**

Karten, die zur alleinigen Darstellung jeweils einer geographischen Einheit einer anderen Karte verwendet werden können, werden als die *untergeordnete Karten* dieser Karte bezeichnet. Einer Karte können dabei eine, mehrere oder auch keine untergeordnete Karte(n) zugeordnet sein; eine Karte kann zugleich für jeweils andere Karten als untergeordnete bzw. als übergeordnete Karte fungieren. So bilden bspw. in einer administrativen Hierarchie

chie die Karten der einzelnen Kontinente die untergeordneten Karten einer Weltkarte sowie die Deutschlandkarte eine untergeordnete Karte der Europakarte; die Europakarte ist damit zugleich eine der untergeordneten Karten der Weltkarte und die übergeordnete Karte der Deutschlandkarte.

▪ Karten

Basierend auf den eingeführten Begriffen kann nun das gewählte Modell zur Beschreibung von Karten definiert werden. Eine *Karte* wird dazu als Entität betrachtet, die folgende Eigenschaften erfüllt:

1. Jeder Karte ist ein eindeutiger Identifikator zugeordnet, anhand dessen auf die Daten der Karte zugegriffen werden kann;
2. Jeder Karte sind ein Name sowie Informationen, die ihre Georeferenzierung erlauben, zugeordnet;
3. Jeder Karte sind Informationen zugeordnet, aus denen abgeleitet werden kann, ob eine ihr übergeordnete Karte existiert und wie der Identifikator dieser übergeordneten Karte lautet;
4. Jede Karte besteht aus einem Satz geographischer Einheiten;
5. Jeder geographischen Einheit ist ein Name zugeordnet;
6. Jeder geographischen Einheit sind Informationen zugeordnet, aus denen abgeleitet werden kann, ob für diese eine untergeordnete Karte existiert und wie der Identifikator dieser untergeordneten Karte lautet;
7. Jeder geographischen Einheit ist ein Satz von Polygonen zugeordnet, die anhand von Vektordaten beschrieben werden.

Da sich jede Karte auf diese Weise selbst in die zugehörige Hierarchie einordnet und die ihr übergeordnete sowie ihre untergeordneten Karte(n) identifiziert, ist es möglich, zur Laufzeit beliebige Kartenhierarchien allein durch das Laden einer zugehörigen Karte aufzuspannen. Da zudem die Information über zugängliche Karten allein in anderen Karten - und damit vollständig außerhalb der Komponente - verankert ist, ist die geforderte Veränderung und Erweiterung von Hierarchien ohne Eingriff in den Programmcode möglich.

17.3.4 Karten unterschiedlicher Detailgenauigkeit

Um das für individuelle Interaktionen jeweils erforderliche Vektordatenvolumen gering zu halten, werden von der Scientific Data Management Group Karten mit jeweils annähernd gleichem Datenvolumen und unterschiedlichem Detaillierungsgrad zur Verfügung gestellt. Diesem Prinzip liegt die Beobachtung zugrunde, dass durch spezifische Nutzerinteraktionen jeweils unterschiedliche Anforderungen an die erforderliche Detailgenauigkeit einer geeigneten Karte gegeben sind. So ist es bspw. für die Auswahl des Kontinentes Afrika hinreichend, eine Karte mit den Umrissen der einzelnen Kontinente darzustellen, während für die Auswahl des Landkreises Teltow-Fläming eine entsprechende Gliederung Brandenburgs zur Verfügung stehen muss. Entsprechend stellen in jeder Hierarchie übergeordnete Karten größere geographische Weltausschnitte mit geringerem Detailgrad dar, untergeordnete Karten hingegen jeweils kleinere Ausschnitte mit höherem Detailgrad. So beschreiben in der verwendeten administrativen Hierarchie bspw. die Vektordaten einer Weltkarte nur die Umrise der Kontinente, während administrative Kontinentkarten die Grenzen von Staaten und administrative Staatenkarten wiederum die Grenzen von Bundesländern oder analogen Subeinheiten beinhalten. Diese Aufteilung ist allerdings nicht apodiktisch – so wurde beispielsweise zusätzlich eine weitere Weltkarte bereitgestellt, die auch Staatsgrenzen enthält. Da IDA beliebig organisierte Karten verarbeiten kann, ist eine flexible Einbindung jeweils anders aufgeteilter Karten für entsprechende Anforderungen unaufwendig möglich.

17.3.5 Dreistufige Datenreduktion

Um ein Datenvolumen für jede Karte zu erreichen, das klein genug für einen performanten Transport über ein Netzwerk ist, wurde von der Scientific Data Management Group ein geeignetes proprietäres Vektordaten-Format entwickelt. Die Vektordaten stammen aus Geoinformationssystemen (GIS) und werden in drei Schritten einer Datenreduktion unterzogen (vgl. Abb. 17.2).

▪ Erste Stufe – Generalisierung

Während die von Geoinformationssystemen verwendeten Vektordaten sehr hohen Anforderungen an Detailgenauigkeit genügen können und daher ein oft großes Datenvolumen aufweisen, genügt für die gegebenen Anforderungen ein geringerer Detailgrad. Entsprechend werden die GIS-Daten zunächst in Abhängigkeit von der geographischen Ausdehnung des von einer Karte abgebildeten Weltausschnittes generalisiert. Dazu werden aus jedem Vektor alle Koordinatenpaare entfernt, deren Abstand zu einem anderen Koordinatenpaar des selben Vektors einen bestimmten Schwellwert unterschreitet.



Abb. 17.2 - Dreistufiger Prozess der Datengewinnung und -reduktion.

▪ Zweite Stufe – Transformation

Vor dem Hintergrund heute gängiger Bildschirmauflösungen wurde pro Karte eine Auflösung von maximal 1000 unterscheidbaren Positionen je Koordinatenachse als hinreichend für die Genauigkeitsanforderungen bezüglich graphischer Darstellung und Interaktion eingestuft. Die Vektoren der GIS-Daten, die zumeist anhand von Paaren geographischer Koordinaten in Form reeller Zahlen repräsentiert werden, werden daher in einen entsprechend kleineren ganzzahligen Wertebereich mit geringerem Speicherplatzbedarf transformiert. Die Rücktransformation dieser Werte in geographische Koordinaten erfolgt zur Laufzeit durch IDA auf dem Rechner des Anwenders.

▪ Dritte Stufe – Aufspaltung

Eine nochmalige deutliche Verringerung des Datenvolumens beruht auf der Beobachtung, dass die Repräsentation der geographischen Einheiten einer Karte durch jeweils geschlossene Polygone eine Vielzahl von Koordinatenpaaren redundant kodiert. Dies ist immer dann gegeben, wenn zwei Polygone über eine gemeinsame Grenzlinie verfügen; die entsprechenden Koordinatenpaare werden dann sowohl zur Beschreibung des einen wie des anderen Polygons - und damit doppelt - gespeichert. Entsprechend findet als dritter Schritt der Datenreduktion eine Aufspaltung der Vektordaten in redundanzfreie *Polylinien* statt²⁷⁰, die so codiert werden, dass die Komponente zur Laufzeit auf dem Rechner des Anwenders eine automatische Rekonstruktion der Polygone durchführen kann.

Durch diese mehrstufige Datenreduktion ist es möglich, das Datenformat so mit Vektordaten anzureichern, dass sämtliche erforderlichen Informationen zur Generierung hinreichend detaillierter Karten in ASCII-Dateien mit einem Datenvolumen von in der Regel jeweils nur rund 10 bis 15 Kilobyte bereitgestellt werden können. Zur schnellen Generierung des Datenformates aus den Ausgangsdaten wurden von der Scientific Data Management Group entsprechende Werkzeuge entwickelt, die es erlauben, diesen Prozess weitgehend automatisch durchzuführen.

²⁷⁰ Auch Geoinformationssysteme verwenden intern eine redundanzfreie Datenhaltung.

17.4 Entwurf der Programmlogik

Nachfolgend wird die gewählte interne Struktur von IDA erläutert. Die Aufgaben der Programmlogik umfassen die Bereitstellung der Karten über die graphische Oberfläche (GUI) der Komponente sowie die Entgegennahme und geeignete Umsetzung der über das GUI ausgelösten Nutzerinteraktionen. Zur Realisierung wurden fünf Module entworfen, die jeweils eine zentrale Aufgabe übernehmen und die erforderlichen Dienste den anderen Modulen zur Verfügung stellen. Auf diese Weise wird eine weitgehende Unabhängigkeit der einzelnen Module voneinander erreicht und somit eine unaufwendige Veränderbarkeit und Erweiterbarkeit der Komponente unterstützt. Die Programmlogik wird gebildet aus Modulen zur internen Repräsentation und generischen Verarbeitung der jeweils aktiven Karte (ActiveMap, Kap. 17.4.1), zur Kartenbereitstellung (MapProvider, Kap. 17.4.2), zur Transformation zwischen unterschiedlichen Koordinatensystemen (TransformationManager, Kap. 17.4.3), zur Steuerung der Visualisierung (Renderer, Kap. 17.4.4) sowie zur Umsetzung der verschiedenen Nutzerinteraktionen (InteractionManager, Kap. 17.4.5). Abb. 17.3 gibt einen Überblick über das Zusammenwirken der einzelnen Module.

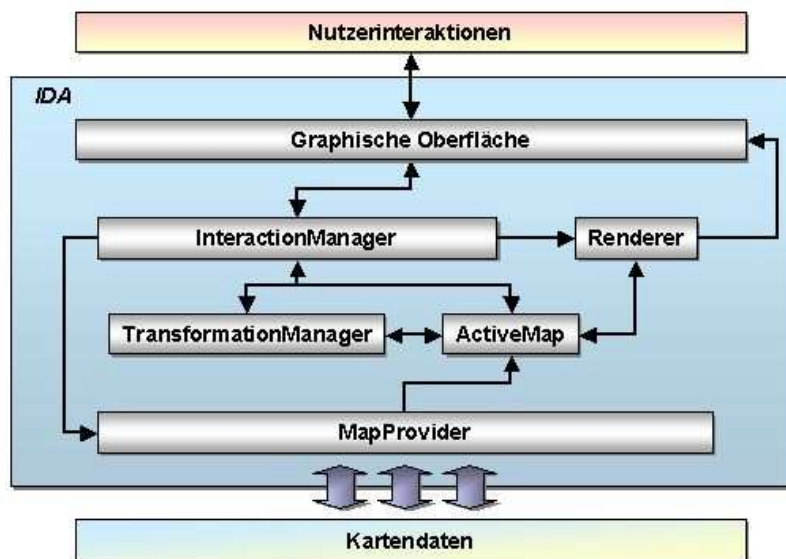


Abb. 17.3 - Aufbau der Programmlogik des Interactive Digital Atlas IDA.

17.4.1 Interne Repräsentation der aktiven Karte

Die von IDA dem Anwender zu einem bestimmten Zeitpunkt zugänglich gemachte Karte wird als *aktive Karte* bezeichnet. Das Modul ActiveMap bildet die Basis für die Erschließung der jeweils aktiven Karte durch die übrigen Module sowie zur generischen Bereitstellung der für Feedback, Navigation und Selektion erforderlichen Informationen in transparenter und komfortabler Form. Entsprechend geht ActiveMap weit über einen passiven Datenspeicher hinaus und stellt vielfältige Dienste zur Verfügung:

- | | |
|------------------------|---|
| Graphische Darstellung | ▶ ActiveMap erzeugt auf Anforderung eine graphische Darstellung der Vektordaten der aktiven Karte für einen gegebenen Pixelraum; |
| Transformationen | ▶ ActiveMap transformiert auf Anforderung die Vektordaten der aktiven Karte in ein gegebenes Koordinatensystem oder extrahiert einen definierbaren Ausschnitt aus der aktiven Karte, um diesen als neue aktive Karte bereitzustellen; |
| Innen-Außen-Relationen | ▶ ActiveMap berechnet auf Anforderung für ein gegebenes Koordinatenpaar, ob sich dieses innerhalb eines der durch die Vektor- |

- daten definierten Polygone der aktiven Karte befindet sowie welcher geographischen Einheit dieses Polygon zugeordnet ist;
- Zustands-
veränderungen
- ▶ ActiveMap selektiert und deselektiert auf Anforderung geographische Einheiten der aktiven Karte anhand von Koordinatenwerten oder Namen.

Ferner stellt ActiveMap auf Anforderung Informationen bereit, die sich auf die aktive Karte als ganzes oder ihre geographischen Einheiten beziehen, und bestimmt die dazu gegebenenfalls erforderlichen Innen-Außen-Relationen selbständig. Zu diesen Informationen zählen beispielsweise:

- Aktive Karte
 - ▶ der Name der aktiven Karte sowie die geographischen Koordinaten der Boundingbox, die die Georeferenzierung dieser Karte beschreibt;
- Übergeordnete Karte
 - ▶ Details zum Zugriff auf die der aktiven Karte übergeordnete Karte (Angaben über Existenz und Identifikator einer solchen Karte);
- Geographische Einheiten
 - ▶ dynamisch berechnete Zusammenhänge zwischen Koordinatenpaaren und den geographischen Einheiten der aktiven Karte (Bestimmung, ob ein gegebenes Koordinatenpaar (x, y) innerhalb einer geographischen Einheit liegt sowie Angaben über den Namen dieser geographischen Einheit);
- Untergeordnete Karten
 - ▶ Details zum Zugriff auf die der aktiven Karte untergeordneten Karten (Angaben über Existenz und Identifikator einer solchen Karte für eine konkrete geographischen Einheit).

Exemplarisch soll hier kurz das zur Umsetzung verwendete Objektmodell skizziert werden, da sich hieran die Vorteile einer objektorientierten Modellierung veranschaulichen lassen. Die erforderliche Funktionalität wird auf generische Weise für jede aktive Karte allein durch das Zusammenspiel von Objekten dreier unterschiedlicher Klassen - bezeichnet als Map, GeoUnit sowie SubShape - bereitgestellt; das gewählte Entwurfsmodell beruht auf einer hierarchischen Aufteilung in die Entitäten Karte, geographische Einheit und Polygon.

- Klasse Map
 - ▶ Die aktive Karte wird vom ActiveMap-Modul anhand eines Objekts der Klasse Map verwaltet. Ein Map-Objekt repräsentiert jeweils eine Karte und verwaltet deren Namen, ihre Georeferenzierung, Informationen über Existenz und Identifikator einer übergeordneten Karte sowie einen Satz von Objekten der Klasse GeoUnit zur Repräsentation der einzelnen geographischen Einheiten dieser Karte.
- Klasse GeoUnit
 - ▶ Jeweils eine geographische Einheit wird durch ein Objekt der Klasse GeoUnit repräsentiert. Jedes GeoUnit-Objekt verwaltet dazu den Namen der geographische Einheit, Informationen über Existenz und Identifikator einer zugehörigen Detailkarte sowie einen Satz von Objekten der Klasse SubShape zur Repräsentation der Vektordaten dieser geographischen Einheit.
- Klasse SubShape
 - ▶ Ein SubShape-Objekt repräsentiert jeweils die Vektordaten eines Polygons und stellt die erforderliche Logik für dessen Transformation, graphische Darstellung sowie für die Berechnung von Innen-Außen-Relationen bereit.

Die Daten einer Karte werden einer hierarchischen Struktur repräsentiert, die generisch

aus Objekten dieser drei Klassen aufgebaut wird (vgl. Abb. 17.4). Dabei übernimmt jede Ebene der Hierarchie spezifische Aufgaben und stellt sie der jeweils nächsthöheren Ebene transparent zur Verfügung. Objekte jeder Ebene erfüllen einen Teil ihrer Aufgaben autonom und reichen alle weiteren Anforderungen jeweils an die Objekte der nächsttieferen Ebene weiter.

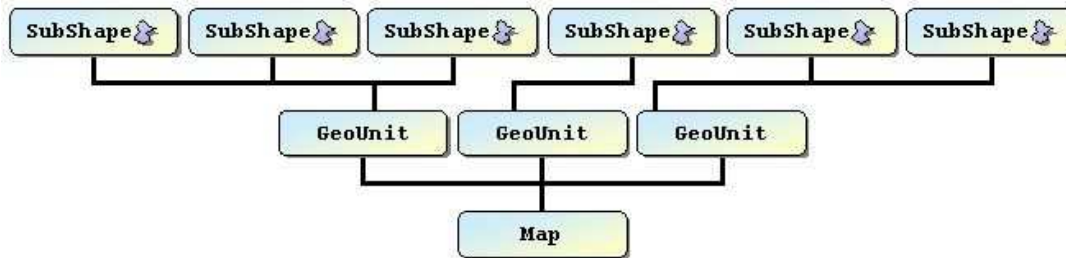


Abb. 17.4 - Schematischer Aufbau der Kartenrepräsentation durch Objekte der Klassen Map, GeoUnit und SubShape.

So kann ein Map-Objekt bspw. selbständig Informationen über eine jeweils übergeordnete Karte bereitstellen, während es für die dynamische Identifikation geographischer Einheiten die von ihm verwalteten GeoUnit-Objekte konsultiert. Soll eine Identifikation geographischer Einheiten anhand von Namen erfolgen, kann jedes GeoUnit-Objekt selbständig agieren, während es für eine Identifikation anhand von Koordinaten, der einen Abgleich mit den Vektordaten erfordert, die von ihm verwalteten SubShape-Objekte heranzieht. Tab. 17.1 gibt einen Überblick über zentrale Funktionalitäten, die über ein Map-Objekt abrufbar sind, sowie über die jeweilige Einbeziehung von GeoUnit- und SubShape-Objekten.

Funktionalität	Beteiligte Objekte		
	Map	GeoUnit	SubShape
Graphische Darstellung der Karte	♦	♦	♦
Transformation der Karte	♦	♦	♦
Selektion einer geographische Einheit über Koordinaten	♦	♦	♦
Selektion einer geographische Einheit über Namen	♦	♦	
Zugriff auf den Namen der Karte	♦		
Zugriff auf die Georeferenzierung der Karte	♦		
Informationen über die übergeordnete Karte	♦		
Zugriff auf den Namen einer geographischen Einheit	♦	♦	
Informationen über der Karte untergeordnete Karten	♦	♦	

Tab. 17.1 - Von Objekten der Klasse Map bereitgestellte Funktionalität (Ausschnitt) und jeweils involvierte Objekte der Klassen GeoUnit und SubShape.

17.4.2 Dynamische Kartenbereitstellung

Die Aufgabe des Moduls MapProvider ist der Austausch der aktiven Karte gegen eine vom Anwender angeforderte neue Karte. Das Modul wird von IDA durch Übergabe des Identifikators der angeforderten Karte aktiviert. Sämtliche Details der Bereitstellung - Zugriff auf die erforderlichen Kartendaten über Netzwerk, lokale Festplatte oder internen Cache, Bereitstellung der angeforderten Karte als Map-Objekt sowie ihre Transformation in das aktuelle Pixelkoordinatensystem der graphischen Oberfläche - werden dabei vom MapProvider gekapselt. Er besteht dazu aus drei interne Komponenten, dem MapCache zur internen Vorhaltung bereits generierter Map-Objekte, dem MapLoader zum Zugriff auf programmexterne Datenressourcen sowie der MapFactory zur Generierung von Map-Objekten aus Kartendaten (vgl. Abb. 17.5).

- **MapCache**

Einmal erzeugte Map-Objekte werden programmintern im MapCache vorgehalten. Er kapselt die Details der Kartenverwaltung und führt eine Abbildung *Kartenidentifikator* → *Map*

Objekt durch. Der MapCache überprüft dazu zunächst anhand des Kartenidentifikators, ob die angeforderte Karte bereits als Objekt vorliegt. Ist dies der Fall, wird das entsprechende Map-Objekt an den MapProvider übergeben; ansonsten übergibt der MapCache den Kartenidentifikator an den MapLoader mit der Aufgabe, die entsprechenden Kartendaten zu laden.

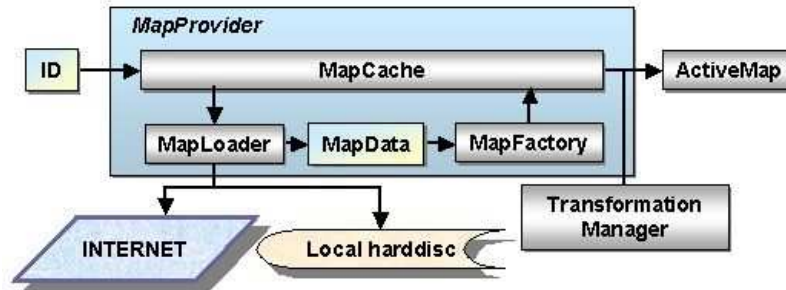


Abb. 17.5 - Transparente Kartenbereitstellung über den MapProvider.

▪ MapLoader

Der MapLoader dient zur Bereitstellung von Karten über ein Netzwerk oder von einer lokalen Festplatte. Er kapselt die Details des entfernten oder lokalen Zugriffs und führt eine Abbildung *Kartenidentifikator* → *Kartendaten* durch. Um die Performance des Programms während eines Ladezugriffs nicht zu beeinträchtigen, wird das Laden asynchron durchgeführt.

▪ MapFactory

Zur Generierung eines neuen Map-Objekts aus den geladenen Kartendaten übergibt der MapLoader diese an die MapFactory, die die Details von Interpretation und Umwandlung kapselt und eine Abbildung *Kartendaten* → *Map-Objekt* durchführt. Ein neu generiertes Map-Objekt wird zum MapCache hinzugefügt.

Nachdem eine Karte entweder direkt im Cache vorgefunden oder zunächst geladen und in diesen eingefügt wurde, wird das entsprechende Map-Objekt aus dem MapCache bereitgestellt. Die Vektordaten der Karte werden in Kommunikation mit dem TransformationManager (vgl. Kap. 17.4.3) entsprechend der aktuellen Ausprägung des Pixelkoordinatensystems transformiert. Abschließend wird das Map-Objekt dem ActiveMap-Modul übergeben und die neue Karte dem Anwender zur Verfügung gestellt.

17.4.3 Transformation zwischen Koordinatensystemen

Die Aufgabe des Moduls TransformationManager ist die transparente Umwandlung von Koordinatenwerten zwischen den jeweils aktuellen Ausprägungen dreier Bezugssysteme. Die drei Systeme sind:

- | | |
|----------------------------------|--|
| Geographisches Koordinatensystem | ▶ Ein <i>geographisches Koordinatensystem</i> zur Repräsentation des von einer Karte dargestellten Weltausschnitts anhand geographischer Koordinaten. Der jeweils aktuelle Wertebereich ist veränderlich und hängt von dem von einer Karte abgebildeten Weltausschnitt ab. Er kann eine maximale Ausdehnung von $\pm 180.0^\circ$ für die X-Achse (geographische Länge, West-Ost-Ausdehnung) und $\pm 90.0^\circ$ für die Y-Achse (geographische Breite, Nord-Süd-Ausdehnung) erreichen. |
| Internes Koordinatensystem | ▶ Ein <i>internes Koordinatensystem</i> zur Vorhaltung der Vektordaten. Der Wertebereich für X- und Y-Werte ist fix und liegt jeweils zwischen 0 und 999. |

- Pixelkoordinatensystem
 - Ein *Pixelkoordinatensystem* zur graphischen Darstellung einer Karte auf dem Bildschirm. Der jeweils aktuelle Wertebereich ist veränderlich und hängt von der über die graphische Oberfläche der Komponente festgelegten aktuellen Darstellungsgröße der Karte ab.

Umwandlungen zwischen den drei Bezugssystemen sind in vielfältigen Formen erforderlich. So müssen bspw. für das Rendering einer Karte die Werte des internen Koordinatensystems in das Pixelkoordinatensystem abgebildet werden; Feedback für das Zeigen auf einen Punkt der graphischen Darstellung erfordert wiederum eine Abbildung vom Pixelkoordinatensystem in das geographische Koordinatensystem etc. Abb. 17.6 zeigt den schematischen Aufbau des TransformationManager-Moduls. Es stellt den übrigen Komponenten die erforderlichen Umwandlungen über entsprechende Schnittstellen bereit, an die Werte jeweils eines Koordinatensystems übergeben werden, um sie in eines der beiden anderen Koordinatensysteme zu transformieren. Dazu verwaltet der TransformationManager Informationen über die aktuellen Ausprägungen der drei Koordinatensysteme, die von den anderen Modulen über entsprechende Schnittstellen aktualisiert werden können, sowie die erforderlichen Projektionsalgorithmen zur Durchführung der Umrechnung. Jede Veränderung eines Bezugssystems - durch Navigation zu einer anderen Karte, durch Zoomen der aktiven Karte, durch Veränderung der Displaygröße - führt zur automatischen Justierung der entsprechenden Parameter, so dass zu jeder Zeit eine korrekte und einheitliche Transformation für alle Module gewährleistet ist.

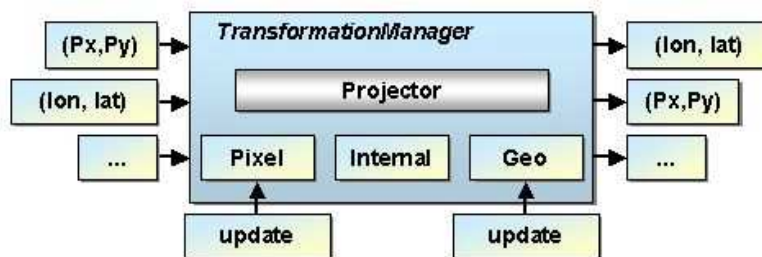


Abb. 17.6 - Transparente Umwandlung von Werten zwischen drei Bezugssystemen durch den TransformationManager.

17.4.4 Visualisierung

Das Renderer-Modul bildet die zentrale Einheit zur Kontrolle und Durchführung der graphischen Darstellung der jeweils aktiven Karte. Eine performante Realisierung des Rendering wird durch Vorberechnung und Vorhaltung von Bildern aus den Vektordaten erreicht. Für die Umsetzung wurde ein Schichtenmodell gewählt, das die sukzessive Überlagerung verschiedener Bildebenen (Layer) bei der Visualisierung erlaubt (vgl. Abb. 17.7).

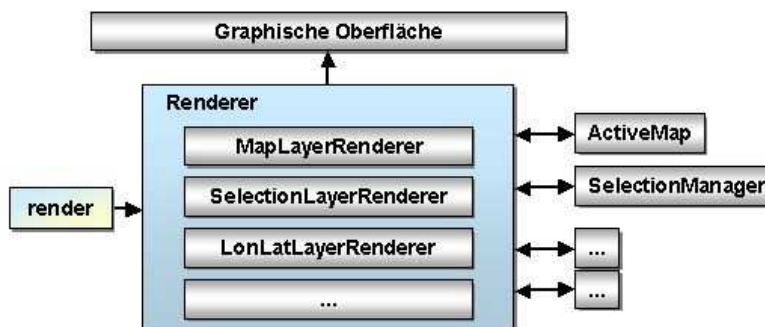


Abb. 17.7 - Der Renderer (Schematischer Aufbau).

Die Ausführung jedes Layers wird durch eine eigene Komponente gesteuert, die gegebene

nenfalls mit anderen Programmbestandteilen kommuniziert. So kontrolliert etwa der MapLayerRenderer die Darstellung der aktiven Karte und löst entsprechende Rendering-Operationen des ActiveMap-Moduls aus, während der SelectionLayerRenderer die graphische Hervorhebung von Selektionen übernimmt und dazu mit dem für deren Verwaltung verantwortlichen SelectionManager (vgl. Kap. 17.4.5) kommuniziert. Die Aufgabe des LonLatLayerRenderer ist die Berechnung und Visualisierung eines dem aktuell dargestellten Weltausschnitt entsprechenden Gradnetzes, das zur Orientierung des Nutzers über Längen- und Breitengrade dient.

Die Aufteilung in voneinander getrennte Schichten, die Visualisierungsaufgaben jeweils autonom sowie in Kommunikation mit anderen Programmbestandteilen erfüllen, erlaubt eine unaufwendige Erweiterbarkeit der Darstellungsmöglichkeiten durch Hinzufügen entsprechender weiterer Layer. Sie bildet damit die Basis für die sukzessive Erweiterung von IDA um Möglichkeiten zur Visualisierung georeferenzierter Daten (vgl. zur Visualisierung von Stationen Kap. 21.2.4 und 25.3 sowie zur prototypischen Visualisierung gegitterter sowie polygonverorteter Zeitreihen Kap. 26.3.3 und 26.3.4).

17.4.5 Umsetzung der Nutzerinteraktionen

Das Modul InteractionManager dient zur Verwaltung und Verteilung der verschiedenen Interaktionen, die vom Anwender über die graphische Oberfläche von IDA ausgelöst werden können. Auftretende Interaktionen werden anhand des für IDA entworfenen Interaktionsmodells (vgl. 17.5.3) identifiziert, klassifiziert und zur spezifischen Weiterverarbeitung an entsprechende Managerkomponenten für Feedback, Selektion, Navigation oder Zooming weitergeleitet. Abb. 17.8 zeigt den schematischen Aufbau des InteractionManager.

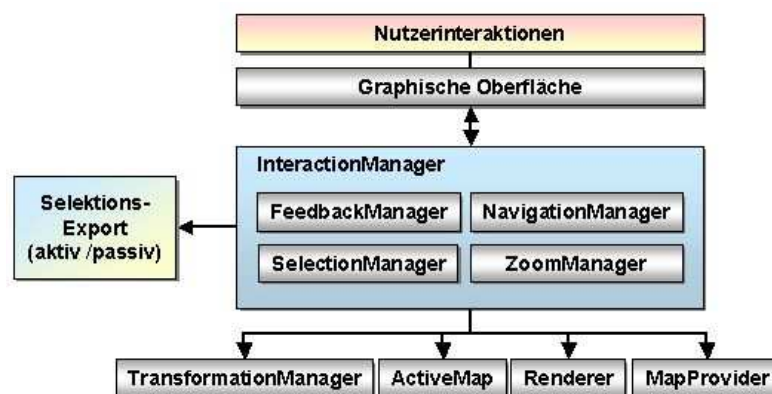


Abb. 17.8 - Schematischer Aufbau des InteractionManager.

▪ FeedbackManager

Die Aufgabe des FeedbackManager ist die Generierung unmittelbarer Rückmeldungen entsprechend des Feedbackmodells von IDA (vgl. Kap. 17.5.2). Die erforderlichen Informationen werden durch Kommunikation mit dem ActiveMap-Modul und dem TransformationManager-Modul identifiziert, aufbereitet und an die graphische Oberfläche weitergeleitet.

▪ SelectionManager

Der SelectionManager übernimmt die Umsetzung von Interaktionen, die zur Selektion oder Deselektion eines Teilraumes führen. Die Nutzerinteraktion wird klassifiziert und in Kommunikation mit dem ActiveMap-Modul umgesetzt. Die Hervorhebung der Selektion wird durch einen entsprechenden Auftrag an den Renderer ausgelöst; die Selektionsparameter in Kommunikation mit dem TransformationManager in geographische Koordinaten transformiert und für einen Export aus der Komponente heraus bereitgestellt. Der externe Zugriff auf die Selektionsparameter kann auf zwei Wegen erfolgen:

- Passiver Export ▶ Externe Komponenten greifen über eine entsprechende Schnittstelle direkt auf die Selektionsparameter zu.
- Aktiver Export ▶ Externe Komponenten registrieren sich über eine entsprechende Schnittstelle für eine automatische Benachrichtigung; der SelectionManager leitet die aktuellen Selektionsparameter weiter, sobald eine Raumauswahl stattgefunden hat.

▪ **NavigationManager**

Der NavigationManager setzt entsprechende Nutzerinteraktionen in die Navigation zu einer anderen Karte um. Durch Kommunikation mit dem ActiveMap-Modul wird identifiziert, zu welcher über- oder untergeordneten Karte navigiert werden soll, ob eine solche verfügbar ist und wie ihr Identifikator lautet. Der Identifikator wird an den MapProvider übergeben, der Kartenbereitstellung und Transformation übernimmt. Abschließend wird der Renderer aufgerufen, um die graphische Darstellung der neuen Karte durchzuführen.

▪ **ZoomManager**

Der ZoomManager steuert die Vergrößerung von Kartenausschnitten (*Zoom in*) sowie den Rücksprung zur Ausgangskarte (*Zoom out*). Dazu wird das Modul ActiveMap aufgefordert, den geforderten Ausschnitt aus seinen Kartendaten zu berechnen und bereitzustellen; das geographische Bezugssystem des TransformationManager-Moduls wird entsprechend aktualisiert. Die graphische Darstellung der veränderten Karte erfolgt abschließend durch Aufruf des Renderer-Moduls.

17.5 Interaktion

Voraussetzung für eine anwenderfreundliche Bereitstellung der Funktionalität sind intuitive und konsistente Metaphern der Bedienung. Um diese zu unterstützen, werden für die Bedienung von IDA die nachfolgend beschriebenen Prinzipien verwendet.

17.5.1 Vollständige Mausbedienbarkeit

Die vollständige Kontrolle der Funktionalität von IDA ist über eine Computermaus möglich. Die geforderten Interaktionen - schnelles Zeigen auf beliebige Punkte der graphischen Darstellung einer Karte, Auswahl von Teilbereichen einer Karte durch Anklicken des entsprechenden Teils der graphischen Darstellung, Markieren eines frei wählbaren rechteckigen Teilausschnitts - sind mit diesem Eingabegerät schnell und intuitiv durchführbar.

17.5.2 Feedbackmodell

Das Feedbackmodell dient zur Definition der automatisch zu generierenden Rückmeldungen bei Interaktionen des Nutzers mit der graphischen Darstellung einer Karte. Folgende Feedbackformen wurden festgelegt:

- Permanentes Feedback ▶ Sobald sich der Mauszeiger über der graphischen Darstellung befindet, werden die der Mauszeigerposition entsprechenden geographischen Koordinaten angezeigt.
- Kontextsensitives Feedback I ▶ Sobald sich der Mauszeiger über der graphischen Darstellung einer geographischen Einheit befindet, wird der Name dieser Einheit angezeigt. Ferner wird ein Hinweis ausgegeben, ob eine Detailkarte für diese verfügbar ist.
- Kontextsensitives Feedback II ▶ Sobald sich der Mauszeiger über der graphischen Darstellung befindet, werden die mit linker und rechter Maustaste jeweils möglichen Aktionen angezeigt.

17.5.3 Interaktionsmodell

Das Interaktionsmodell der Komponente dient zur Definition einer vollständigen und konsistenten Mausbedienung. Folgende Interaktionsformen wurden festgelegt:

- | | |
|--------------------------------------|---|
| Zeigen | ▶ Das Zeigen auf einen beliebigen Punkt der Karte dient zur Auslösung des interaktiven Feedback entsprechend des Feedbackmodells (vgl. Kap. 17.5.2). Zeigen erfolgt durch Bewegen des Mauszeigers über der graphischen Darstellung der aktuellen Karte. |
| Gummiband | ▶ Das Gummiband dient zur freien Raumauswahl oder zur Definition eines Bereiches, der aufgezoomt werden soll. Das Aufziehen des Gummibandes erfolgt mit gedrückter <i>linker</i> Maustaste über der graphischen Darstellung der aktuellen Karte. |
| Selektion geographischer Einheiten | ▶ Die Auswahl geographischer Einheiten der aktuellen Karte erfolgt durch Zeigen auf ihre jeweilige graphische Darstellung und Betätigung der <i>linken</i> Maustaste. |
| Selektion eines Kartenausschnitts | ▶ Die Auswahl eines frei wählbaren rechteckigen Ausschnitts der aktuellen Karte erfolgt durch Aufziehen des Gummibandes auf der graphischen Darstellung der Karte. |
| Navigation zu untergeordneten Karten | ▶ Die Navigation zur Detailkarte einer geographischen Einheit erfolgt durch Zeigen auf ihre jeweilige graphische Darstellung und Betätigung der <i>rechten</i> Maustaste. |
| Navigation zu übergeordneten Karten | ▶ Die Navigation zu der der aktuellen Karte übergeordneten Karte erfolgt durch Zeigen auf eine beliebige Position außerhalb der Kartenpolygone und Betätigung der <i>rechten</i> Maustaste. |
| Navigation über Menu | ▶ Zusätzlich kann direkt zu jeder beliebigen Karte jeder eingebundenen Hierarchie durch Anwahl eines entsprechenden Eintrages aus einem Menubaum navigiert werden. |
| Zoom in | ▶ Das Vergrößern eines nutzerdefinierbaren Teilausschnittes der aktuellen Karte erfolgt durch Aufziehen des Gummibandes, ein nachfolgendes Zeigen an eine beliebige Position innerhalb des so markierten Bereiches und die Betätigung der <i>rechten</i> Maustaste. |
| Zoom out | ▶ Der Rücksprung aus einem aufgezoomten Teilausschnitt zur Ausgangskarte erfolgt analog zur Navigation zu einer übergeordneten Karte durch Zeigen auf eine beliebige Position außerhalb der Kartenpolygone und Betätigung der <i>rechten</i> Maustaste. |

Auf einer aufgezoomten Karte stehen dem Anwender die selben Interaktionen zur Verfügung wie auf nichtgezoomten Karten. Neben dem Zugriff auf maussensitives Feedback über geographische Koordinaten oder dargestellte Gebiete besteht damit die Möglichkeit, einen Ausschnitt einer bereits aufgezoomten Karte sukzessive weiter zu vergrößern. Dabei wird gegenwärtig eine bis zu 70fache Vergrößerung relativ zur Ausgangskarte unterstützt. Im Gegensatz zur Navigation zu anderen Karten (vgl. Abb. 17.9a bis 17.9c) wird eine aufgezoomte Karte von IDA dynamisch aus der jeweiligen Ausgangskarte generiert, indem die durch das Gummiband definierte Boundingbox ausgewertet und eine entsprechende Transformation durchgeführt wird (vgl. Kap. 17.4.5 sowie Abb. 17.9d bis 17.9f).

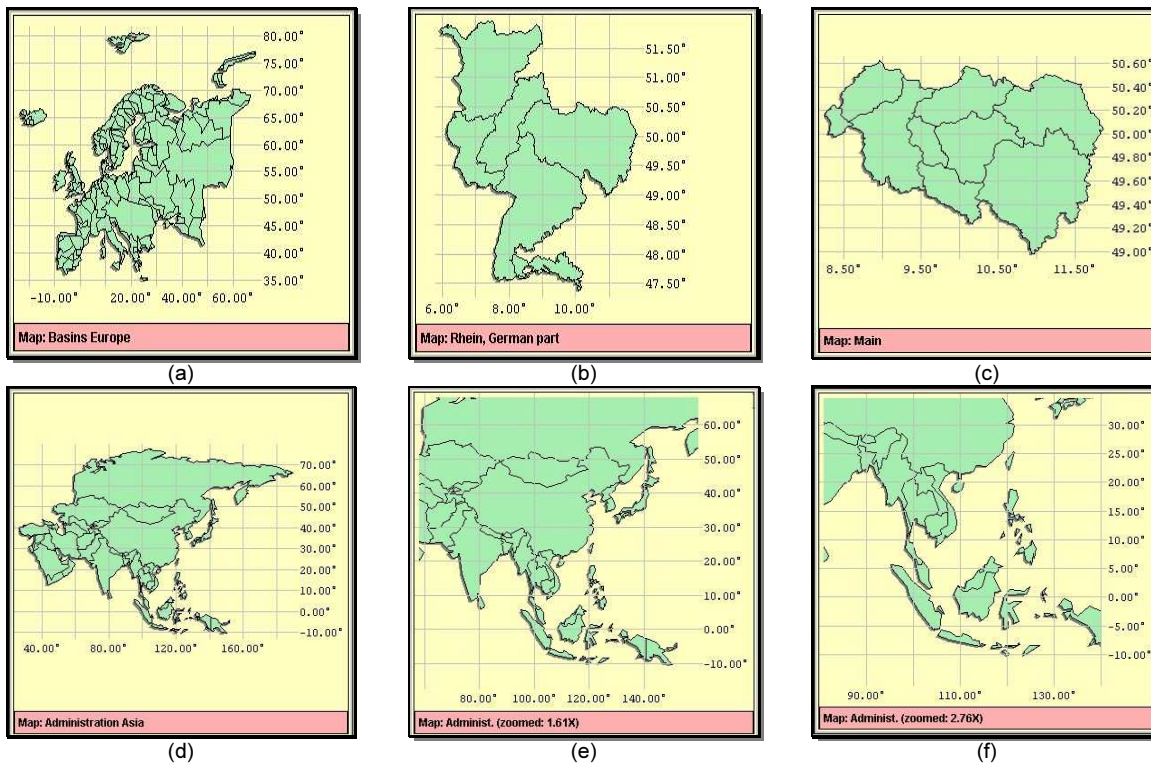


Abb. 17.9 - Beispiele für Navigation (obere Reihe) und Zooming (untere Reihe). Navigation zu detaillierteren Flusseinzugsgebietskarten: (a) Europa, (b) Rhein, deutscher Teil; (c) Main; nutzergesteuerte sukzessive Vergrößerung einer administrativen Asienkarte: (d) Ausgangskarte; (e) und (f) interaktiv erzeugte Ergebniskarten durch Auswahl von Teilbereichen und Aufzoomen.

17.5.4 Konfigurierbare Raumauswahl

Für die flexible und intuitive Auswahl von Raumbezügen stellt die Komponente gegenwärtig fünf unterschiedliche Möglichkeiten zur Auswahl, die extern konfiguriert werden können. Diese werden abschließend kurz beschrieben:

- **Freie Selektion einer Boundingbox**

Die freie Auswahl einer Boundingbox dient zur nutzerdefinierten Selektion eines beliebigen, rechteckigen Ausschnittes auf der aktuell dargestellten Karte, der durch vier geographische Koordinaten beschrieben wird, die den Ecken des Rechtecks entsprechen. Der Ausschnitt kann auf jeder Karte durch Aufziehen des Gummibandes bestimmt werden (vgl. Abb. 17.10a).

- **Selektion einer geographischen Einheit**

Die Selektion einer geographischen Einheit erfolgt durch direkte Anwahl ihrer graphischen Darstellung auf der aktuellen Karte. Sämtliche zugehörigen Polygone werden farblich hervorgehoben (vgl. Abb. 17.10b).

- **Selektion der Boundingbox einer geographischen Einheit**

Ist die Komponente für die Selektion boundingbox-basierter Raumbezüge konfiguriert, führt die Anwahl einer geographischen Einheit zur automatischen Selektion der entsprechenden Boundingbox. Die Polygone der geographischen Einheit werden farblich hervorgehoben; zusätzlich wird die zugehörige Boundingbox graphisch dargestellt (vgl. Abb. 17.10c).

- **Selektion mehrerer geographischer Einheiten**

Die Selektion einer nutzerdefinierten Kombination geographischer Einheiten der aktuell dargestellten Karte erfolgt durch sequentielle Anwahl einzelner Einheiten. Die zugehörigen Polygone der ausgewählten geographischen Einheiten werden farblich hervorgehoben (vgl. Abb. 17.10d). Alternativ zu einer sequentiellen Anwahl kann eine nutzerdefinierte Kombi-

nation geographischer Einheiten durch Aufziehen eines Gummibandes bestimmt werden; in diesem Fall werden alle geographischen Einheiten automatisch selektiert, deren graphische Darstellung ganz oder teilweise in der so definierten Boundingbox liegt (vgl. Abb. 17.10e).

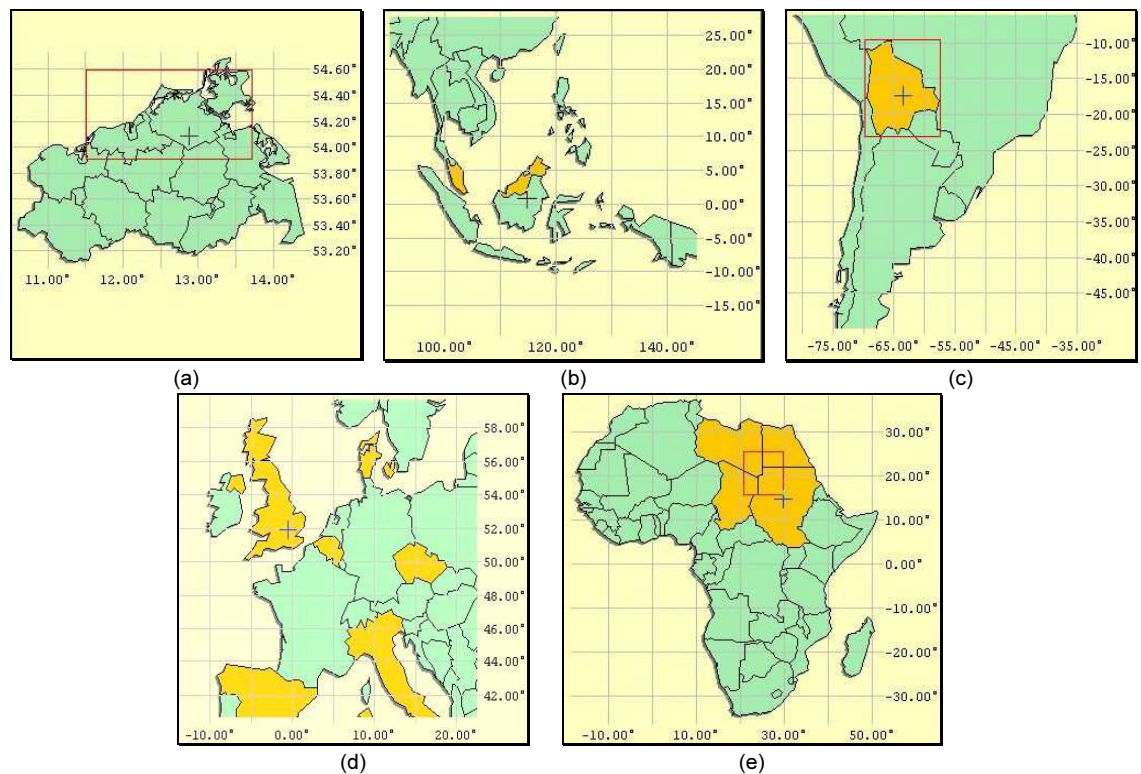


Abb. 17.10 - Beispiele für unterschiedliche Formen der Raumselektion: (a) Selektion eines Teilgebietes von Mecklenburg-Vorpommern; (b) Auswahl von Malaysia; (c) Auswahl der Boundingbox von Bolivien; (d) Auswahl verschiedener europäischer Staaten durch schrittweise Auswahl; (e) Auswahl von Libyen, Ägypten, Tschad und Sudan durch Gummiband.

Zur Selektion von Raumbezügen wurde IDA in das Filtermodul SpatialFilter eingebettet (vgl. Kap. 20.3). Ferner findet die Komponente Verwendung bei der Visualisierung von Zeitreihenmetadaten über das Auswertungsmodul StationVisualizer (vgl. Kap. 21.2.4); sie wurde zudem für verschiedene weitere Aufgaben nachgenutzt (vgl. Kap. 25.1, 25.3 und 26.3).

