

# TEIL C

## LÖSUNGSSTRATEGIEN UND KONZEPTION DER SCHNITTSTELLE

In Teil C wird die gewählte Konzeption zur Realisierung einer intuitiven und funktionalen Schnittstelle zur Erschließung der allgemeinen Metadaten, Zeitreihenmetadaten sowie punktverorteten Zeitreihen des Institutes dargestellt. Zunächst werden die eingesetzten Lösungsstrategien beschrieben (Kap. 11) und die grundlegenden Systemanforderungen zusammengefasst (Kap. 12). Die Darstellung des Softwareentwurfs gliedert sich in die Beschreibung von Grobkonzept (Kap. 13), Feinkonzept (Kap. 14), Programmlogik (Kap. 15) sowie der Verteilung auf Client und Server (Kap. 16). Abschließend wird die Konzeption der Komponente IDA (Interactive Digital Atlas) beschrieben, die zur komfortablen Selektion von Raumbezügen sowie zur raumbezogenen Visualisierung von Ergebnisdaten entwickelt wurde (Kap. 17).

## 11 Eingesetzte Lösungsstrategien

In diesem Kapitel werden die eingesetzten Lösungsstrategien beschrieben. In einem einführenden Teil (Kap. 11.1) werden zunächst Aspekte aufgeführt, die bei der Realisierung von Nutzerschnittstellen zu beachten sind. Die gewählten Strategien zur Erzielung einer geeigneten Lösung gliedern sich in einen iterativen Entwicklungsprozess unter Verwendung von Prototypen (Kap. 11.2), die Unterstützung des Anwenders durch Möglichkeiten zur graphisch-interaktiven Selektion von Raumbezügen ebenso wie zur interaktiven Visualisierung selektierter Stationen und Zeitreihen (Kap. 11.3), die Reduzierung der Bedienungskomplexität (Kap. 11.4) sowie Flexibilität und Adaptivität der Software (Kap. 11.5).

### 11.1 Einführung

In *The Psychology of Human-Computer Interaction* stufen Card, Moran und Newell die Interaktion von Anwender und Computer als einen kommunikativen Dialog<sup>228</sup> ein, der dem Zweck dient, *Aufgaben* erfolgreich zu bewältigen [Card et al. 1983, 4]. Geeignete Nutzeroberflächen besitzen dabei zentrale Bedeutung für die intuitive Bedienbarkeit und Akzeptanz von Software-Anwendungen (vgl. bspw. [Shneiderman 1998]); ihre Gestaltung wird jedoch häufig unterschätzt [Balzert 1999, 655].

Der eigentliche Zweck einer Software - und damit auch der zu entwickelnden Schnittstelle - ist es, ihren Anwendern die Umsetzung ihrer Intentionen in einer Weise zu ermöglichen, die den jeweiligen Anforderungen und Erwartungshaltungen entspricht. Gelingt dies nicht, besteht die Gefahr, dass die Software als ungeeignet empfunden und letztlich abgelehnt wird. Dazu müssen sowohl die gegebenen technologischen Möglichkeiten wie die Anforderungen und Bedürfnisse ihrer Anwender verstanden und miteinander in Einklang gebracht werden:

- ▶ „*Harnessing the computer's power is a task für designers who understand the technology and are sensitive to human capacities and needs.*“ [Shneiderman 1998, 4]

Aus dem Forschungsgebiet der *Human Computer Interaction* (HCI) (auch bezeichnet als *Computer Human Interaction*, CHI) sind fünf Kriterien bekannt, die als Anhaltspunkte für eine Qualitätsbeurteilung von Nutzerschnittstellen herangezogen werden können [Shneiderman 1998, 15f.]:

- |                                  |   |
|----------------------------------|---|
| Lerndauer                        | ▶ Wie lange brauchen typische Anwender, um die Ausführung bestimmter Aufgaben mit der Software zu erlernen?                                   |
| Ausführungsgeschwindigkeit       | ▶ Wie lange dauert es, mit der Software jeweils bestimmte Aufgaben auszuführen?   |
| Bedienungsfehler                 | ▶ Wie hoch ist die Zahl der Fehler, die bei der Benutzung der Software gemacht werden, und um welche Arten von Fehlern handelt es sich dabei? |
| Wiederabrufbarkeit des Erlernten | ▶ Wie gut bewahren Anwender ihr Wissen über die Bedienung der Software nach Ablauf von Stunden, Tagen oder Wochen?                            |
| Subjektive Befriedigung          | ▶ Wie gut gefällt Anwendern die Benutzung der verschiedenen Aspekte der Software?   |

Die Gestaltung von Schnittstellen, die diese Kriterien möglichst gut erfüllen, wird insbeson-

---

<sup>228</sup> Die Autoren sprechen hier von einem Dialog, da beide Parteien Zugriff auf einen zur Kommunikation ausgetauschten Strom von Symbolen haben und an verschiedenen Punkten aktiv in die Kommunikation eingreifen können.

dere dadurch erschwert, dass hierbei eine Vielzahl von menschlichen Faktoren zu berücksichtigen sind. Shneiderman verweist ferner darauf, dass in der Praxis Wechselwirkungen zwischen diesen Kriterien bestehen – so kann eine lange Lerndauer zu erhöhter Ausführungsgeschwindigkeit beitragen, während die Sicherstellung einer extrem niedrigen Rate von Bedienungsfehlern hingegen zu Lasten der Ausführungsgeschwindigkeit gehen kann. Ebenso können je nach Anwendungsgebiet jeweils andere Kriterien - etwa subjektive Befriedigung oder kurze Lerndauer - als wichtigstes Ziel einzustufen sein.

### 11.1.1 Unterschiede zwischen Menschen

Shneiderman betont die große Bedeutung der vielfältigen Unterschiede zwischen einzelnen Menschen. Zu beachten sind etwa [Shneiderman 1998, 18ff.]:

- |  |  |
|--|--|
| Körperliche Fähigkeiten                    | ▶ Hier sind beispielsweise unterschiedliche Reichweiten im Sitzen, unterschiedliche Geschwindigkeit der Finger, mögliche Farbenblindheit etc. zu berücksichtigen.  |
| Kognitive und perzeptuelle Fähigkeiten     | ▶ Einzubeziehen sind u.a. unterschiedliche Ausprägungen von Kurz- und Langzeitgedächtnis, individuelle Ausprägungen der Lernfähigkeit, Unterschiede im Umgang mit Problemlösungen und Entscheidungsfindung, unterschiedliche Grade von Aufmerksamkeit, unterschiedliche Wahrnehmungen von Zeit etc.  |
| Persönliche Vorlieben und Unterschiede     | ▶ Ferner ist zu beachten, dass Menschen sowohl in ihrem Verhältnis zu Computern (Antipathie oder Sympathie, diesbezügliche Angst oder Attraktion) sowie in ihren Persönlichkeitsmerkmalen unterschiedlich sind. So ist beispielsweise zwischen extrovertierten und introvertierten Menschen, gefühls- oder verstandesbetonten Menschen, aber auch zwischen Frauen und Männern <sup>229</sup> zu unterscheiden.   |
| Kulturelle und internationale Unterschiede | ▶ Je nach Anwendungshorizont einer Software dürfen auch die vielfältigen Unterschiede zwischen einzelnen Kulturen oder Ländern nicht außer acht gelassen werden. Nicht vernachlässigt werden dürfen hier beispielsweise unterschiedliche Datums- und Zeitformate, numerische Formate, Währungsformate, Anreden und Titel, die jeweilige Bedeutung von Farben und Metaphern oder auch unterschiedliche Formen von Lesen und Eingabe (von links nach rechts, von rechts nach links oder vertikal). |

Besondere Voraussetzungen gelten zudem für Software für ältere Benutzer mit nachlassenden Fähigkeiten beispielsweise bezüglich Sehen, Hören, Geschwindigkeit, für blinde Benutzer, für Kinder mit Lernbehinderungen etc.

### 11.1.2 Unterschiede in Nutzergruppen

Weitere Herausforderungen entstehen dadurch, dass in der Regel nicht von einem homogenen Nutzertypus ausgegangen werden kann. [Shneiderman 1998, 68ff.] unterscheidet drei

---

<sup>229</sup> [Shneiderman 1998, 21ff.] geht davon aus, dass hier fundamentale Unterschiede bestehen, stellt jedoch fest, dass keine klaren Muster für Vorlieben bekannt sind. So wisse man, dass unter frühen Videospiele von Anwenderinnen primär *Pacman* sowie einige andere Spiele wie *Donkey Kong* oder *Tetris* bevorzugt wurden, könne über die Ursachen hierfür allerdings nur spekulieren. Er verweist ferner darauf, dass im Kontext von Office-Systemen die zumeist männlichen Programmdesigner möglicherweise nicht an die Wirkung auf weibliche Anwender gedacht haben, als sie Befehle wie „*KILL a file*“ oder „*ABORT a program*“ einführten (*Abortion* bedeutet im Englischen *Abtreibung*, aber auch *Fehl-, Früh- oder Missgeburt*).

Gruppen von Nutzern mit jeweils unterschiedlichen Anforderungen und Bedürfnissen und betrachtet diese jeweils im Bezug auf voraussetzbare Kenntnisse hinsichtlich der mit einer Software auszuführenden Aufgabe sowie hinsichtlich der Konzepte von Schnittstellen.

- |                                  |  |
|----------------------------------|--|
| Novice or first-time users       | ▶ Bei (echten) Anfängern ist von geringen Kenntnissen sowohl bezüglich der Aufgabe wie der Schnittstellenkonzepte auszugehen; erstmalige Benutzer werden hingegen als erfahren in der Aufgabe eingestuft, besitzen jedoch nur oberflächliche Kenntnisse der Schnittstellenkonzepte. Eine geeignete Schnittstelle soll solche Benutzer dabei unterstützen, sowohl Berührungspunkte abzulegen wie Nutzerwissen aufzubauen.   |
| Knowledgeable intermittent users | ▶ Zu dieser Gruppe werden kenntnisreiche (knowledgeable) Anwender gerechnet, die sporadisch (intermittent) eine Vielzahl unterschiedlicher Systeme nutzen. Solche Benutzer werden als sowohl erfahren in der auszuführenden Aufgabe wie in Schnittstellenkonzepten eingestuft; allerdings ist davon auszugehen, dass sie Schwierigkeiten haben, sich an die Details der jeweiligen Bedienung einer Software zu erinnern. Eine geeignete Schnittstelle soll solche Benutzer entsprechend bei der Vergewärtigung von Details der Bedienung unterstützen. |
| Expert frequent users            | ▶ Bei Anwendern dieser Benutzergruppe wird davon ausgegangen, dass sie sowohl mit der auszuführenden Aufgabe wie den Schnittstellenkonzepten gründlich vertraut sind und zudem Wert darauf legen, ihre Arbeit schnell zu erledigen. Entsprechend soll eine geeignete Schnittstelle solchen Benutzern Bedienungsmöglichkeiten bereitstellen, die eine schnelle und effiziente Erledigung ihrer Aufgaben unterstützen.   |

Diese Charakteristika sind nach Shneiderman für jede Anwendung jeweils zu verfeinern; die Anforderungen an den Entwurf einer Schnittstelle wachsen entsprechend mit der Zahl unterschiedlicher Nutzergruppen, deren Bedürfnisse gleichermaßen zufriedengestellt werden müssen.

### 11.1.3 Faktoren für die Schnittstellengestaltung

Die Aspekte der Interaktion zwischen Menschen und Computern wurden von diversen Autoren anhand verschiedener Modelle und aus unterschiedlichen Blickwinkeln untersucht (vgl. bspw. die Modelle GOMS und das Keystroke-Level Model von [Card et al. 1983], das Modell CE+ von [Polson, Lewis 1990] sowie die Übersicht bei [Shneiderman 1998, 51ff.]). Basierend auf den Ergebnissen wurden Regeln vorgeschlagen, die zur Entwicklung besser geeigneter Schnittstellen beitragen sollen (vgl. bspw. die *Design rules based on Analyses of Human Error* von [Norman 1983], die *Seven Principles for Transforming Difficult Tasks into Simple Ones* von [Norman 1988, 188ff.], die acht *Design Principles for Successful Guessing* von [Polson, Lewis 1990] oder *The Eight Golden Rules of Interface Design* von [Shneiderman 1998, 74ff.]). Es lassen sich u.a. folgende Faktoren festhalten, die besser beherrschbare Schnittstellen auszumachen scheinen:

- |                       |   |
|-----------------------|---|
| Gefühl der Kontrolle  | ▶ Es ist ein Design zu bevorzugen, das dem Anwender ein Gefühl der Kontrolle über die Software vermittelt.                                  |
| Erforderliches Wissen | ▶ Es ist ein Design zu bevorzugen, das die Menge und die Komplexität des zur Bedienung der Software erforderlichen neuen Wissens minimiert. |

- Gedächtnisbelastung ▶ Die Belastung des Kurzzeitgedächtnisses des Anwenders soll gering gehalten werden. Um dies zu erreichen, sollen bspw. übersichtliche Formen der Darstellung verwendet und Online-Zugriffe auf zur Bedienung erforderliche Informationen unterstützt werden.
- Konsistenz ▶ Es ist ein möglichst hoher Grad an Konsistenz sicherzustellen. Dies umfasst bspw. die Bereitstellung konsistenter Aktionen für einander ähnliche Situationen, die Verwendung einer identischer Terminologie in verschiedenen Teilen der Schnittstelle sowie Einheitlichkeit in Layout, Farben, Schriftarten, Groß- und Kleinschreibung etc.
- Feedback ▶ Für jede Aktion des Anwenders sind geeignete und informative Rückmeldungen (*Feedback*) bereitzustellen.
- Umkehrbarkeit ▶ Es ist so weit wie möglich eine einfache Umkehrbarkeit von Aktionen des Nutzers zu ermöglichen, um eine angstfreie Erforschung der Software zu unterstützen.
- Vermeidung von Fehlern ▶ Der Anwender ist so weit wie möglich vor schwerwiegenden Bedienungsfehlern zu bewahren.
- Behandlung von Fehlern ▶ Es soll eine einfache Fehlerbehandlung unterstützt werden; Fehler des Anwenders sollen erkannt und einfache, konstruktive Vorschläge zur Behebung angeboten werden.

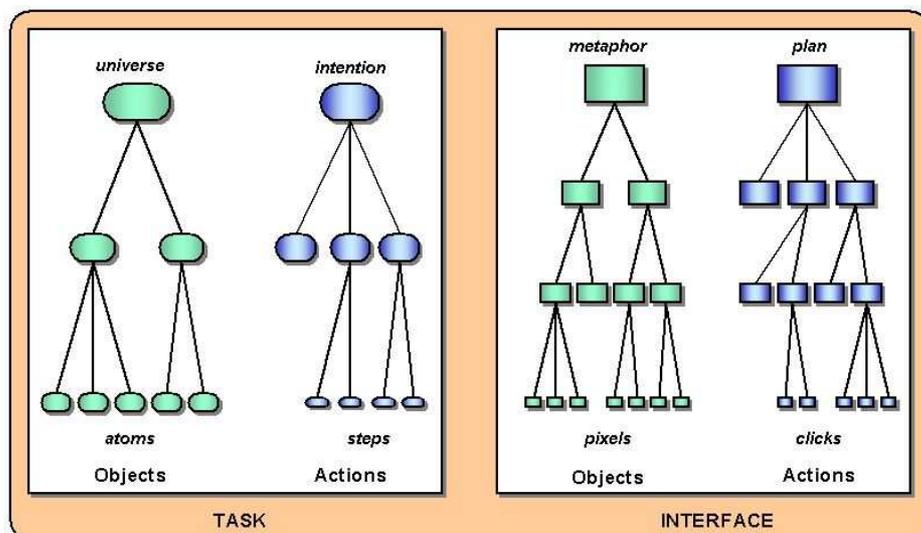


Abb. 11.1 - Das Object-Action Interface Model: Aufteilung von Aufgaben und Schnittstelle in Hierarchien von Objekten und Aktionen<sup>230</sup>.

#### 11.1.4 Das Object-Action Interface Model

Eine allgemein anwendbare „Bauanleitung“, deren Befolgung die Gestaltung jeweils optimal geeigneter Schnittstellen zwischen beliebigen Anwendern und Computern für unterschiedlichste Aufgaben ermöglichen würde, existiert bis heute nicht. [Shneiderman 1998, 61ff.] schlägt das *Object-Action Interface Model* (OAI) vor, das der Tatsache Rechnung trägt, dass graphische Nutzeroberflächen zunehmend die Interaktion des Anwenders über Kommandosprachen ersetzen. Das Object-Action Interface Model legt den Schwerpunkt

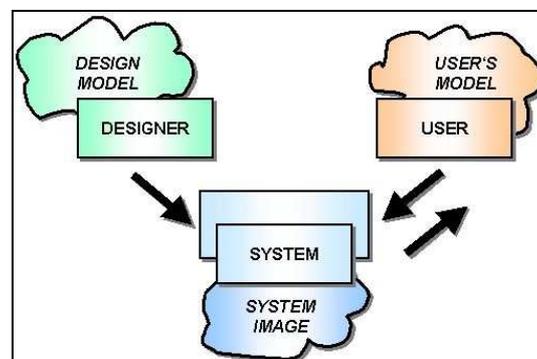
<sup>230</sup> Abbildung erstellt in Anlehnung an [Shneiderman 1998, 62, Fig. 2.2].

auf die Aufspaltung der vom Nutzer zu bewältigenden Aufgaben in *Hierarchien* aus einzelnen Objekten und Aktionen sowie deren geeignete Abbildung durch ebenfalls hierarchisch strukturierte Objekte und Aktionen einer Schnittstelle (vgl. Abb. 11.1).

Auf diese Weise kann ein komplexes Problem so lange hierarchisch in kleinere Einheiten zerlegt werden, bis jedes Teilproblem handhabbar ist<sup>231</sup>. Shneiderman konstatiert, dass Hierarchien zwar nicht perfekt, aber verständlich und nützlich seien und dass die meisten Nutzer eine Aufspaltung ihrer Aufgaben in Objekte und Aktionen auf höheren und tieferen Ebenen akzeptieren. Er schlägt vor, das Object-Action Interface Model für den Entwurf von interaktiven Systemen anzuwenden, um diese Arbeit zu systematisieren: Dazu sollen zunächst die Objekte einer Aufgabe identifiziert und die entsprechenden Aktionen möglichst klar beschrieben werden. Darauf aufbauend können nachfolgend die entsprechenden Objekte und Aktionen der zu entwickelnden Schnittstelle identifiziert und in geeignete Repräsentationen umgesetzt werden.

### 11.1.5 Mentale Modelle

Von entscheidender Bedeutung für intuitiv bedienbare Software - wie für Werkzeuge generell - ist das innere Bild, das sich der Anwender von der möglichen Art der Interaktion macht. Donald Norman, der in *The Psychology Of Everyday Things* untersucht, warum manche Gegenstände einfach zu verstehen und zu benutzen sind, während andere ihre Anwender immer wieder vor Probleme stellen, weist in diesem Zusammenhang auf die zentrale Rolle *mentaler Modelle (Mental Models)* hin [Norman 1988]. Diese stellen nach Norman die Modelle dar, die Leute von sich selbst, von anderen, der Umgebung sowie den Dingen haben, mit den sie interagieren; sie werden basierend auf dem jeweils verfügbaren Wissen entwickelt. Das mentale Modell, das ein Anwender von einem Gegenstand entwickelt, wird dabei zu großen Teilen durch Interpretation seiner sichtbaren Struktur sowie der wahrgenommenen Aktionen geprägt.



**Abb. 11.2** - Die Rolle konzeptueller Modelle bei der indirekten Kommunikation zwischen Designer und Anwender eines Systems.<sup>232</sup>

Bedeutsam ist, dass der Designer eines Systems und dessen Anwender nicht direkt miteinander kommunizieren, sondern nur indirekt durch das System selbst. Norman zufolge sind dabei drei unterschiedliche Aspekte mentaler Modelle zu unterscheiden, die auch als *konzeptuelle Modelle (Conceptual Models)*<sup>233</sup> bezeichnet werden. Er führt dazu die Begriffe

<sup>231</sup> Dies ist dem klassischen Ansatz Descartes' aus dem Jahr 1637 nicht unähnlich, der sich als eine seiner Regeln auferlegte: „[...] jedes Problem, das ich untersuchen würde, in so viele Teile zu teilen, wie es angeht und wie es nötig ist, um es leichter zu lösen“ (vgl. René Descartes, *Von der Methode des richtigen Vernunftgebrauchs und der wissenschaftlichen Forschung*. Hamburg, Felix Meiner Philosophische Bibliothek Band 26a, unveränderter Nachdruck 1978, 15).

<sup>232</sup> Abbildung erstellt nach [Norman 1988, 190, Fig. 7.1].

<sup>233</sup> Ein geeignetes konzeptuelles Modell erlaubt die Vorhersage der Auswirkungen, die durch spezifische Aktionen ausgelöst werden [Norman 1988, 13].

Design Model, System Image und User's Model ein (vgl. Abb. 11.2). Das *Design Model* ist die Vorstellung des Designers eines Systems, wie dieses zu funktionieren hat. Das *System Image* entsteht aus der physikalischen Struktur eines Systems sowie seiner Dokumentation, Anweisungen und Beschriftungen. Das *User's Model* schließlich ist das mentale Modell, das vom Anwender durch Interaktion mit dem System entwickelt wird. Ein System wird nach Norman dann problemlos zu bedienen sein, wenn User's Model und Design Model äquivalent sind. Wenn jedoch das System Image das Design Model nicht klar und konsistent vermittelt, wird der Nutzer ein falsches mentales Modell von der Funktionsweise des Systems entwickeln.

### 11.1.6 Das unabdingbare Verständnis für die Anwender

Shneiderman betont, dass es für die Entwicklung geeigneter Nutzerschnittstellen wesentlich ist, die Anwender und ihre Aufgaben in den Mittelpunkt aller Bemühungen zu stellen. Entsprechend fordert er:

- ▶ „*Concentrate attention on the users and on the tasks that they must accomplish. Make users the center of attention and build feelings of competence, mastery, clarity, and predictability.*“ [Shneiderman 1998, 596]

Unabdingbare Voraussetzung hierfür ist es, zu einem angemessenen Verständnis der jeweiligen Bedürfnisse der Anwender zu gelangen. Zum Abschluss sollen hier zwei Kommentare zur diesbezüglichen Forderung von Wilfried J. Hansen aufgeführt werden, der 1971 in seinen *User Engineering Principles for Interactive Systems* formulierte:

- ▶ „*Know thy user*“<sup>234</sup>

Shneiderman äußert sich in diesem Kontext mehr als zwanzig Jahre darauf bereits deutlich vorsichtiger:

- ▶ „*The process of getting to know the users is never ending because there is so much to know and because the users keep changing.*“ [Shneiderman 1998, 68]

Radikaler fällt hingegen der diesbezügliche Kommentar von Bertrand Meyer aus. Meyer, einer der Vordenker im Bereich der objektorientierten Programmierung, empfiehlt in seinen Anmerkungen zum Einsatz dieses Konzeptes für die Gestaltung von Nutzerschnittstellen, so wenig wie möglich bezüglich der adressierten Anwender vorauszusetzen. Meyers *User Interface Design principle* lautet daher:

- ▶ „*Do not pretend you know the user; you don't.*“ [Meyer 1997, 12]

Auch für die zu realisierende Schnittstelle mussten Wege gefunden werden, die adressierte komplexe Gruppe der Anwender und ihre Bedürfnisse hinreichend zu verstehen, um darauf aufbauend ein System entwickeln zu können, das den Anwendern wiederum die Ableitung eines möglichst geeigneten mentalen Modells seiner Bedienung erlaubt. Um hierbei die Gefahr von Fehlentwicklungen zu minimieren sowie die als unabdingbar eingestufte Akzeptanz der Schnittstelle durch die Gruppen der Datennutzer und Datenbereitsteller zu erlangen, wurde auf mehrere ineinandergreifende Lösungsstrategien zurückgegriffen, die nachfolgend beschrieben werden.

---

<sup>234</sup> Zitiert nach [Shneiderman 1998, 67]. Meyer gibt Hansen in leicht unterschiedlicher Form mit „*know the user*“ wieder [Meyer 1997, 12]. Die bei beiden Autoren angegebene Quelle ist [Hansen 1971].

## 11.2 Iterativer Entwicklungsprozess

Im Bereich des Software-Engineering wurden Prozessmodelle entwickelt, um den Herausforderungen zu begegnen, die sich bei der Produktion von Software ergeben (vgl. hierzu bspw. [Hennings 1983]). Für die Realisierung der Schnittstelle erschien dabei der Rückgriff auf traditionelle Prozessmodelle zur Softwareentwicklung wie Wasserfall-Modell oder V-Modell<sup>235</sup>, die eine vollständige Spezifikation der Anforderungen zu Beginn der Entwicklung voraussetzen, wenig erfolversprechend. Solche Modelle sind nach [Balzert 1998, 114] u.a. dann nicht geeignet, wenn folgende Faktoren bei einer Softwareentwicklung gegeben sind:

- ▶ Auftraggeber und Anwender können ihre Anforderungen nicht explizit und / oder nicht vollständig formulieren;
- ▶ es ist ein kontinuierliches und wechselseitiges Lernen von Anwendern und Entwicklern während der Erstellung der Software erforderlich;
- ▶ für gegebene Anforderungen können unterschiedliche Lösungsmöglichkeiten existieren, die experimentell erprobt und diskutiert werden müssen;
- ▶ die Auftraggeber müssen zunächst von der prinzipiellen Durchführbarkeit überzeugt werden.

Alle vier Herausforderungen trafen auf den gegebenen Kontext zu<sup>236</sup>. Entsprechend wurde für die Entwicklung der Schnittstelle ein *iterativer Prozess* gewählt, der sich sowohl an Konzepten des inkrementellen Prozessmodells (vgl. Kap. 11.2.1) wie des Prototypen-Prozessmodells (vgl. Kap. 11.2.2) orientierte.

### 11.2.1 Evolutionäres und inkrementelles Prozessmodell

Das *evolutionäre Prozessmodell* [Balzert 1998, 120ff.] konzentriert sich auf die Bereitstellung jeweils lauffähiger Teilprodukte und geht dazu zunächst nur von den Kernanforderungen des Auftraggebers aus. Nur diese werden zunächst entworfen und implementiert und in einer sog. Nullversion bereitgestellt. Basierend auf den Erfahrungen des Auftraggebers mit dieser Nullversion werden die Anforderungen für eine erweiterte Version ermittelt, umgesetzt und in einer zweiten Version bereitgestellt, die die Basis für wieder neue Anforderungen bildet. Dieses evolutionäre Vorgehen birgt allerdings die Gefahr, dass durch Konzentration auf die Kernanforderungen wesentliche Anforderungen zu spät erkannt und dadurch nachfolgend aufwendige Umgestaltungen des Systems erforderlich werden. Balzert unterscheidet daher explizit das *inkrementelle Prozessmodell* [Balzert 1998, 122f.], das ebenfalls auf der sukzessiven Entwicklung von Teillösungen basiert und die Einbeziehung mit diesen gewonnenen Erkenntnisse in die nächsten Ausbaustufen erlaubt, allerdings die Vermeidung von Fehlentwicklungen durch eine möglichst vollständige Analyse *aller* Anforderungen vor Beginn der Entwicklung anstrebt.

Der iterative Entwicklungsprozess der Schnittstelle wurde am inkrementellen Prozessmodell ausgerichtet. Da eine vollständige Definition aller Anforderungen nicht möglich war (vgl. Kap. 9.3), wurde angestrebt, vorab die Kernanforderungen so zu definieren, dass eine Anpassung der Software an sich ergebende neue Anforderungen ohne gravierende Umgestaltungen möglich war.

---

<sup>235</sup> Für eine ausführliche Beschreibung und vergleichende Betrachtung unterschiedlicher Prozessmodelle sowie weiterführende Literaturangaben vgl. [Balzert 1998, 97ff.].

<sup>236</sup> Das PIK fungierte zwar als offizieller Auftraggeber, trat in dieser Rolle jedoch letztlich nur stellvertretend für die Vielzahl der Datennutzer und Datenbereitsteller auf, die für die geplante Entwicklung gewonnen werden mussten.

### 11.2.2 Prototypen-Prozessmodell

Um die unabdingbare Einbeziehung verschiedener Anwender in den Entwicklungsprozess zu ermöglichen, wurden zusätzlich Elemente des *Prototypen-Prozessmodells* [Balzert 1998, 114ff.] verwendet. Dieses Modell unterstützt die frühzeitige Erstellung ablauffähiger Softwaremodelle (*Prototypen*), anhand derer gewählte Umsetzungen demonstriert und mit Anwendern erörtert werden können. Dabei können unterschiedliche Arten von Prototypen unterschieden werden. *Demonstrationsprototypen* dienen dazu, Auftraggebern einen ersten Eindruck davon zu vermitteln, wie das spätere Produkt im Prinzip aussehen kann; *Prototypen im engeren Sinne* sind hingegen provisorische, ablauffähige Softwaresysteme, die entwickelt werden, um den Anwendungsbereich und spezifische Aspekte der Funktionalität zu analysieren. Als *Labormuster* werden wiederum solche Prototypen bezeichnet, die zur Klärung technischer Alternativen dienen; sie werden zumeist ohne Einbeziehung der Endbenutzer evaluiert. *Pilotsysteme* schließlich dienen nicht nur zur Veranschaulichung oder Erprobung, sondern bilden zugleich den Kern der zu erstellenden Software und werden zyklisch weiterentwickelt; ihre Entwicklung verläuft dabei ähnlich wie bei einer evolutionären Vorgehensweise (sog. *evolutionäres prototyping*) [Balzert 1998, 120f.].

Für die Entwicklung der Schnittstelle wurden unterschiedliche Formen von Prototypen eingesetzt. Der Systemkern sowie übergreifend verwendete Bestandteile der Schnittstelle wurden als Pilotsysteme konzipiert und iterativ weiterentwickelt; zur Klärung spezifischer Anwenderaspekte wurden gegebenenfalls Demonstrationsprototypen oder Prototypen im engeren Sinne eingesetzt. Um zu Einsichten über eine geeignete Gestaltung der Schnittstelle zu gelangen, ohne die einzubeziehenden Akteure zu sehr bei ihren eigentlichen Aufgaben zu stören, wurde eine fortlaufende informelle Bewertung durch individuelle Datenbereitsteller und Datennutzer verfolgt. Zusätzlich wurden im Bedarfsfall Labormuster entwickelt, um Implementations-Alternativen unter Performance-Aspekten bewerten zu können.

### 11.2.3 Iterative Herausbildung einer zugänglichen Datenschicht

Vor dem gegebenen Hintergrund heterogener Datenräume war es erforderlich, den potentiellen Datennutzern einen möglichst homogenen Zugang zu den Daten anzubieten. Zugleich konnten jedoch die vorhandenen Datenräume der Datenbereitsteller nicht umgestaltet werden, um sie zu homogenisieren, da auf ihrer konkreten Ausprägung eine Vielzahl laufender Anwendungen basiert (vgl. Kap. 9.2.3). Eine Migration der Daten (vgl. Kap. 2.4.3) zur Überwindung ihrer Heterogenität kam damit von vorneherein nicht in Frage. Da zudem keine klaren Vorgaben über eine geeignete Form der Datenzusammenführung gemacht werden konnten (vgl. Kap. 9.3), erschien es ferner unwahrscheinlich, eine Lösung „auf einen Schlag“ ausschließlich durch Einsatz einer der in Kap. 2 vorgestellten virtuellen bzw. materiellen Integrationsstrategien - die überdies zu Beginn der Arbeit z.T. noch nicht zur Verfügung standen - erreichen zu können.

Daher wurde - neben der zu entwickelnden Anwendungssoftware - zugleich das sukzessive Entstehen einer von dieser erschließbaren Datenschicht neben den eigentlichen Ausgangsdaten angestrebt. Die Schnittstelle wurde dazu so konzipiert, dass eine schnelle Anbindung unterschiedlicher Datenbanken möglich ist, die dem Anwender in Form einzeln adressierbarer Datenräume offeriert werden. Über die Schnittstelle sollte entsprechend ein einheitlicher Einsprungspunkt für sämtliche eingebundenen Datenräume bereitgestellt und zunächst die Selektion *jeweils eines Datenraumes* erlaubt werden, um nachfolgend gegen diesen Anfragen durchzuführen. Auf eine systemseitige Integration der eingebundenen Datenräume wurde bewusst verzichtet, um die hierzu erforderlichen Prozesse auf die Ebene der Datenbereitstellung zu verlagern. Auf diese Weise bestand die Möglichkeit, zunächst mit der Bereitstellung ausgewählter Datenbanken des Institutes zu beginnen, die

Zahl der zugänglichen Datenräume schrittweise zu erweitern und schließlich zur Einbindung integrierter Datenräume, die von der Scientific Data Management Group erstellt wurden, überzugehen (zu den hierfür vorgesehenen Entwicklungszyklen vgl. Kap. 11.2.5).

#### 11.2.4 Die Schnittstelle als Katalysator

Im gewählten iterativen Prozess wurde der zu entwickelnden Schnittstelle in mehrfacher Weise die Rolle eines *Katalysators* für sukzessive Verbesserungen und Erweiterungen sowohl von Software wie Datenschicht zugeordnet:

- **Katalysator für Anwenderfeedback**

Die Schnittstelle sollte in jedem Entwicklungszyklus als Grundlage für Erfahrungen individueller Anwender mit dieser dienen. Auf diese Weise gewonnene Erkenntnisse und Anregungen für mögliche Verbesserungen und Erweiterungen sollten in den Entwicklungsprozess zurückfließen und dort entsprechend umgesetzt werden, um so eine schrittweise Annäherung sowohl an die erforderliche Funktionalität wie an ihre geeignete und intuitive Bereitstellung zu erreichen.

- **Katalysator für die Erweiterung der Datenschicht**

Die Schnittstelle sollte in jedem Entwicklungszyklus zur direkten *Demonstration der Vorteile* dienen, die diese beim Zugriff auf die jeweils eingebundenen Datenbanken im Vergleich zu den bisherigen Zugriffswegen bieten kann. Auf diese Weise sollte neben ihrer Etablierung bei einer wachsenden Anzahl von Datennutzern zugleich eine steigende Attraktivität für die Gruppe der Datenbereitsteller erreicht und so eine fortlaufende Erweiterung der jeweils zugänglichen Datengrundlage motiviert werden.

- **Katalysator für die Optimierung der Datenschicht**

Schließlich sollte die Schnittstelle in jedem Entwicklungszyklus zu Erkenntnissen bezüglich einer Optimierung der für diese bereitgestellten Datenschicht führen und so zum Entstehen eines stetig besser erschließbaren Datengrundlage beitragen. Die Schnittstelle sollte anhand ihres jeweiligen Ist-Zustandes zu Erkenntnissen der Beteiligten darüber beitragen, inwiefern eine direkter Mehr an Nutzen für die Anwender - bspw. ein funktionalerer Zugang zu den Daten oder ihre integrierte Erschließung - durch *Veränderungen auf Ebene der Datenbereitstellung* erreicht werden kann.

#### 11.2.5 Vorgesehene Entwicklungszyklen

Der gewählte iterative Entwicklungsprozess zielte sowohl auf die Erstellung der erforderlichen Software wie auf das sukzessive Entstehen der über diese zugänglichen Datenschicht ab. Dabei wurde eine gleichzeitige Annäherung an alle einzubeziehenden Datenräume aufgrund ihrer Komplexität und Heterogenität als wenig geeignet eingestuft. Entsprechend wurde ein mehrstufiger Prozess gewählt:

- **Erster Zyklus – Entwicklung der Kernfunktionalität**

Als Ziel des ersten Zyklus wurde die Entwicklung der erforderlichen Kernfunktionalität für eine Erschließung der allgemeinen Metadaten sowie der Zeitreihenmetadaten des Institutes festgelegt. Konzeption und Realisierung der hierfür erforderlichen Software erfolgten anhand von ausgewählten Datenbanken des PIK. Einbezogen wurden hierfür die allgemeine Metadatenbank CERA-2 sowie eine Datenbank des PIK-Kernprojektes RAGTIME, die als Referenzdatenbank für die Erschließung von Zeitreihenmetadaten verwendet wurde. Beide Datenbanken wurden von verschiedenen Wissenschaftlern administriert, die jedoch jeweils der Scientific Data Management Group zugeordnet sind, so dass ein regelmäßiger und unkomplizierter Austausch über Anforderungen und Eignung von Teillösungen stattfinden konnte. Zur Sicherstellung der Übertragbarkeit auf weitere Zeitreihenmetadaten wurde zusätzlich die globale Meteorologiedatenbank des Institutes einbezogen, die

von der Abteilung Climate System aufgebaut wurde und beständig erweitert wird. In diesem Zyklus wurden Wissenschaftler des Institutes auf informeller Basis einbezogen, um zu Abschätzungen über die Qualität und Akzeptanz der entwickelten Teillösungen zu gelangen und diese entsprechend zu verbessern und weiterzuentwickeln.

▪ **Zweiter Zyklus (Betriebsphase I) – Metadatenzugriff**

Als Ziel des zweiten Zyklus wurde die - prototypische - erste Betriebsphase der Schnittstelle und ihre Etablierung bei den Anwendern anhand des Zugriffs auf die allgemeinen Metadaten und Zeitreihenmetadaten des Institutes vorgegeben. Als konkrete Ziele wurden festgelegt:

- |   |   |
|---|---|
| Bereitstellung der Referenzdatenbanken  | ▶ Zunächst waren die Referenzdatenbanken über die Schnittstelle allen Wissenschaftlern des Institutes zur Verfügung zu stellen. Es wurde erwartet, dass der bis dahin erreichte Stand den grundsätzlichen Nutzen der Lösung überzeugend demonstriert. |
| Sukzessive Erweiterung der Datenschicht | ▶ In der Folge sollte im laufenden Betrieb so lange eine schrittweise Einbindung weiterer Zeitreihenmetadaten erfolgen, bis sämtliche diesbezüglichen Datenräume des Institutes über die Schnittstelle zugänglich sind.                               |

▪ **Dritter Zyklus (laufender Betrieb) – Datenintegration und Ausbau der Software**

Parallel zum zweiten Zyklus sollte, basierend auf den bis dahin gewonnenen Erfahrungen, im laufenden Betrieb die nächste und endgültige Betriebsphase vorbereitet werden:

- |   |  |
|---|--|
| Vorbereitung einer integrierten Datenerschließung             | ▶ Die Scientific Data Management Group sollte in dieser Zeit mehrere Datenbanken zur integrierten Bereitstellung von Zeitreihenmetadaten sowie eine Datenbank zur integrierten Bereitstellung sämtlicher relevanten punktverorteten Zeitreihen des Institutes entwickeln. Parallel dazu sollte die Schnittstelle um die für einen komfortablen Zugang zu den Zeitreihen erforderliche Funktionalität erweitert werden. |
| Ableitung einer allgemeinen Institutsmetadatenbank aus CERA-2 | ▶ Ferner sollte von der Scientific Data Management Group basierend auf den bisherigen Erfahrungen der Anwender mit CERA-2 eine für die Anforderungen des Institutes geeignete Untermenge aus dieser Datenbank abgeleitet werden (PIK CERA-2), die eine effiziente Bereitstellung der für das Informationsbedürfnis der Datennutzer relevanten Attribute ermöglicht.  |

▪ **Vierter Zyklus (Betriebsphase II) – Integrierter Metadaten- und Datenzugriff**

Als Ziele des vierten und letzten Zyklus, der die zweite Betriebsphase der Schnittstelle eröffnet, wurden neben dem Zugriff auf die - nun über PIK CERA-2 bereitzustellenden - allgemeinen Metadaten der integrierte Zugriff auf Zeitreihenmetadaten und Zeitreihen festgelegt. Kann dies in geeigneter Weise umgesetzt und erfolgreich bei den Anwendern etabliert werden, ist das avisierte Ziel ihrer autonomen und individuellen Versorgung aus diesen wesentlichen Datenressourcen des Institutes erreicht.

## 11.3 Unterstützung des Anwenders

Der kombinierte Einsatz jeweils geeigneter graphischer Bedienelemente<sup>237</sup>, ebenso wie entsprechende syntaktische und semantische Eingabeüberprüfungen und geeignete Rückmeldungen an den Anwender können heute als nahezu selbstverständliche Anforderungen eingestuft werden und wurden auch für die Schnittstelle umgesetzt. Ferner wurde besonderer Wert darauf gelegt, den Anwender durch komfortable Möglichkeiten zur flexiblen Selektion von Raumbezügen auf Karten sowie zur interaktiven Visualisierung von Stationen und Zeitreihen zu unterstützen.

### 11.3.1 Graphisch-interaktive Selektion von Raumbezügen

Im gegebenen Anwendungskontext kommt dem in Metadaten und Daten dokumentierten *Raumbezug* eine zentrale Bedeutung zu. Solche sog. Georeferenzierungen treten in den zu erschließenden Datenräumen in verschiedenen Ausprägungen auf, die nachfolgend kurz dargestellt werden:

- |   |  |
|---|--|
| Geographische Extrema (Boundingboxen)   | ▶ Die Beschreibung eines Raumbezuges anhand geographischer Extrema, d.h. durch die Angabe von minimaler und maximaler geographische Breite (engl. Latitude) sowie minimaler und maximaler geographische Länge (engl. Longitude), wird häufig in allgemeinen räumlichen Metadaten verwendet. Solche sog. <i>Boundingboxen</i> , die durch vier geographische Koordinaten definiert sind, werden auch in der allgemeinen Metadatenbank CERA-2 eingesetzt, um die jeweilige räumliche Abdeckung der dokumentierten Daten zu beschreiben.              |
| Koordinatenpaare                        | ▶ Die Georeferenzierung jeder einzelnen dokumentierten Station - und damit zugleich der dieser zugeordneten Zeitreihen - erfolgt in den hier betrachteten Zeitreihenmetadaten anhand eines Paares geographischer Koordinaten, die jeweils die exakte Position der Station definieren (Punktverortung).   |
| Hierarchische räumliche Klassifikatoren | ▶ In einzelnen Datenräumen können Stationen <i>zusätzlich</i> anhand von hierarchischen räumlichen Klassifikatoren einem oder mehreren vordefinierten geographischen Räumen zugeordnet werden, bspw. zu administrativen und / oder naturräumlichen Einheiten (vgl. Kap. 18.3.1). Während die Georeferenzierung anhand von Koordinatenpaaren für jede Station in den betrachteten Datenräumen gegeben ist, ist ihre zusätzliche Georeferenzierung anhand von Klassifikatoren als sich über die Zeit entwickelnder, dynamischer Prozess einzustufen. |

Es war davon auszugehen, dass die Auswahl geographischer Bezüge für die individuellen Datennutzer eine zentrale Rolle bei der Selektion von Untermengen aus den einzelnen Datenräumen spielen wird. Entsprechend wurde beschlossen, eine komfortable und flexible Auswahl von Untermengen sowohl aus allgemeinen Metadaten wie aus Zeitreihenmetadaten durch Möglichkeiten einer graphisch-interaktiven Raumselektion auf Karten zu unterstützen.

---

<sup>237</sup> Bspw. sog. Checkboxes für Parameter, die eine freie Auswahl aus einer Anzahl von Optionen erlauben, Klappboxen (sog. Choices) oder Listen für die Auswahl eines Wertes aus einer Anzahl von Optionen, Textfelder für freie Dateneingabe etc.

### 11.3.2 Interaktive Visualisierung selektierter Stationen und Zeitreihen

Die Visualisierung von Daten (vgl. Kap. 4) stellt ein wichtiges Werkzeug zu deren Auswertung dar und wird entsprechend auch im Kontext der Erdsystemanalyse in vielfältiger Weise eingesetzt. Die avisierten Anwender sind daher mit bestimmten visuellen Repräsentationen bei der Arbeit mit den zu erschließenden Daten vertraut. Um die Attraktivität und damit die Akzeptanz der Schnittstelle zu erhöhen, wurde beschlossen, die Anwender bei der Auswertung zuvor selektierter Ergebnisdaten durch geeignete Formen ihrer interaktiver Visualisierung zu unterstützen. Zu nennen sind insbesondere:

- **Interaktive Visualisierung von Stationen auf Karten**

Da Zeitreihenmetadaten die Position von Stationen in einem räumlichen Bezugssystem definieren, sind sie als raumbezogene Daten aufzufassen, für die entsprechende Visualisierungstechniken herangezogen werden können (vgl. Kap. 4.7.2). Im gegebenen Anwendungskontext werden Zeitreihenmetadaten typischerweise anhand von Geoinformationssystemen auf computergenerierten Karten, die die Georeferenzierung der einzelnen Stationen visualisieren, ausgewertet. Entsprechend wurde beschlossen, Möglichkeiten zur interaktiven raumbezogenen Visualisierung zuvor selektierter Stationen auf Karten bereitzustellen.

- **Interaktive Visualisierung von Zeitreihen**

Für einen intuitiven ersten Eindruck und Vergleich einzelner Zeitreihen - also zeitabhängiger Daten (vgl. Kap. 4.7.1) - werden im gegebenen Anwendungskontext bspw. Linien- oder Verbunddiagramme (vgl. Kap. 4.5.4 und 4.5.6) eingesetzt. Daher wurde beschlossen, eine entsprechende interaktive Zeitreihenvisualisierung auch anhand der Schnittstelle für zuvor vom Anwender selektierte Zeitreihen zu ermöglichen.

Hierbei stand jeweils eine einfach beherrschbare und intuitiv verständliche Visualisierung der Daten im Vordergrund, die zudem mit vertretbarem Entwicklungsaufwand realisierbar sein musste. Auf die Umsetzung spezieller Techniken zur Visualisierung multivariater Daten (vgl. Kap. 4.6) oder von Konzepten der Informationsvisualisierung (vgl. Kap. 4.8) wurde daher im Rahmen dieser Arbeit verzichtet. Da die Schnittstelle jedoch so entworfen wurde, dass eine unaufwendige Erweiterung der Software unterstützt wird (vgl. Kap. 14), können solche und andere weiterführende Konzepte zur Datenauswertung - etwa Techniken des Data Mining (vgl. Kap. 3), die aufgrund ihrer speziellen Aufgabenstellung hier ebenfalls keine Berücksichtigung fanden - bei Bedarf nachträglich integriert werden.

## 11.4 Reduzierung der Bedienungskomplexität

Um die Komplexität der Datenerschließung zu reduzieren, wurde - in gewisser Anlehnung an das Object-Action-Interface-Modell von Shneiderman (vgl. Kap. 11.1.4) - eine hierarchische Aufgliederung der Gesamtaufgabe in kleinere, vom Anwender leichter erlern- und beherrschbare Bestandteile verfolgt. Hierzu wurde die Entwicklung wiederverwendbarer graphischer Nutzerschnittstellen für einzelne, in verschiedenen Datenräumen wiederkehrende Aspekte von Anfrageformulierung und Auswertung von Ergebnisdaten verfolgt.

### 11.4.1 Modulare Datenselektion

Es waren geeignete Formen zur nutzerdefinierten Selektion von Untermengen aus einem zuvor vom Anwender ausgewählten Datenraum bereitzustellen. Dazu waren vielfältige heterogene Selektionskriterien zu unterstützen, die dem Anwender zugänglich zu machen waren und jeweils die Entwicklung geeigneter Nutzerschnittstellen erforderten. Dabei war zu berücksichtigen, dass die Zusammensetzung der zu unterstützenden Selektionskriterien je nach ausgewähltem Datenraum variieren kann. Ebenso konnte nicht davon ausgegangen werden, dass alle Anwender jeweils auf die gleichen Selektionskriterien zur Auswahl

von Daten aus demselben Datenraum zurückgreifen wollen, sondern vielmehr individuell zusammenstellbare Kombinationen bevorzugen. Die gewählte Vorgehensweise basierte daher auf der iterativen Entwicklung wiederverwendbarer und flexibel miteinander kombinierbarer graphischer Nutzerschnittstellen, die dem Anwender jeweils in intuitiver und funktionaler Weise die Definition von Teilen seiner Gesamtanfrage gegen einen zuvor ausgewählten Datenraum ermöglichen sollen.

#### 11.4.2 Modulare Auswertung von Selektionsergebnissen

Für die Entwicklung geeigneter Möglichkeiten zur effizienten Auswertung von zuvor selektierten Metadaten und Daten waren ebenfalls besondere Lösungsstrategien erforderlich. Die gegebene Multidimensionalität der Daten erschwert eine intuitive Auswertung aller in einer Ergebnismenge enthaltenen Informationen und erfordert die Bereitstellung geeigneter graphischer Nutzerschnittstellen, die einen flexiblen Zugang zu diesen eröffnen. Dabei erlauben beispielsweise textuelle Darstellungen den exakten Abruf konkreter Wertausprägungen, bieten zugleich jedoch wenig Unterstützung bei einer intuitiven Auswertung spezifischer Eigenschaften wie der räumlichen Verteilung selektierter Stationen oder des Werteverlauf spezifischer Zeitreihen. Entsprechende Visualisierungen wiederum können dies in geeigneter Weise leisten, dabei jedoch in der Regel jeweils nur einen Ausschnitt aller in den Daten enthaltenen Informationen darstellen. Eine effiziente Datenauswertung kann hier also nur durch die *Kombination* unterschiedlicher Darstellungsformen erreicht werden.

Ferner war zu berücksichtigen, dass nicht jede Form der Darstellung gleichermaßen für jeden im gegebenen Kontext zu adressierenden Datenraum geeignet ist. Dies gilt sowohl für textuelle wie visuelle Formen der Darstellung. So kann je nach Kontext eine tabellarische Darstellung der Wertausprägungen möglichst vieler Ergebnisdatensätze zugleich zu bevorzugen sein, um so deren Vergleich durch den Anwender zu unterstützen, oder - im Falle besonders umfangreicher Informationen, wie sie bspw. über CERA-2 bereitgestellt werden - eine Darstellung, die jeweils die übersichtliche Präsentation einzelner Datensätze leistet. Auch an eine jeweils geeignete Visualisierung waren durch die unterschiedlichen Ausprägungen der Daten jeweils eigene Anforderungen zu stellen. So können bspw. durchaus sehr viele Stationen zugleich auf einer Karte dargestellt werden, während eine gleichzeitige Visualisierung von allgemeinen Metadaten, deren Raumbezug über Boundingboxen, also rechteckigen Flächen, beschrieben ist, auf ein und derselben Karte sehr schnell unübersichtlich wird; die Visualisierung von Zeitreihen wiederum erfolgt sinnvoll anhand geeigneter Diagramme. Die hier gewählte Vorgehensweise basierte daher auf der iterativen Entwicklung wiederverwendbarer und flexibel miteinander kombinierbarer graphischer Nutzerschnittstellen, die dem Anwender jeweils spezifische Formen der Darstellung der Ergebnisdaten und der Interaktion mit diesen ermöglichen sollen.

#### 11.4.3 Vorteile

Dieser Ansatz bietet im gegebenen Kontext mehrere Vorteile, die nachfolgend beschrieben werden.

Komplexitätsreduktion

- ▶ Die Bereitstellung einzelner Nutzerschnittstellen für Teilaufgaben der Datenselektion erlaubt es, komplexe Anfragen hierarchisch in verschiedene, einzeln adressierbare semantische Aspekte zu unterteilen. Auf diese Weise wird für die Anwender die Komplexität der Gesamtaufgabe Datenselektion durch ihre Aufspaltung in kleinere, jeweils einfacher beherrschbare Aufgaben reduziert. Ebenso ermöglicht es eine kombinierte Bereitstellung unterschiedlicher Nutzerschnittstellen für die Datenauswertung den Anwendern, sich der Komplexität eines Anfrageergebnisses aus

unterschiedlichen Blickwinkeln zu nähern und unterstützt so ebenfalls eine Reduzierung der Gesamtkomplexität.

- Konsistenz
- ▶ Die wiederholte Verwendung einzelner Nutzerschnittstellen für Anfragedefinition und Ergebnisauswertung unterstützt die Anwender bei der Benutzung der Schnittstelle in zweifacher Hinsicht. Sie erlaubt zunächst die Übertragung einmal erlernter Funktionalität auf andere Aufgaben, die so zunehmend intuitiv ausgeführt werden können; zudem ermöglicht sie es, anhand vertrauter Elemente einen leichteren Zugang zur Orientierung über individuellen Datenräume trotz Heterogenität und Komplexität zu finden.
- Individualität
- ▶ Da bei der gegebenen Vielzahl von möglichen Vorlieben und Bedürfnissen nicht davon ausgegangen werden kann, dass alle Anwender ihre Anfragen in gleicher Weise formulieren wollen, ist es wesentlich, flexible Formen der Anfrageformulierung zu ermöglichen. Miteinander kombinierbare Nutzerschnittstellen für die Anfragedefinition unterstützen hier die Anwender dabei, sich ihren individuellen Weg zu den Daten auszuwählen. Ferner erlaubt die kombinierte Bereitstellung unterschiedlicher Nutzerschnittstellen zur Ergebnisauswertung den Anwendern, sich entweder auf die von ihnen jeweils bevorzugte Form der Darstellung zu konzentrieren oder die angebotenen Darstellungsformen kombiniert einzusetzen. Dies unterstützt die erforderlichen individuellen Prozesse der Datenauswertung.
- Vielfalt
- ▶ Ziel der Schnittstelle ist es, einen möglichst funktionalen und intuitiven Zugang zu komplexen Datenräumen zu eröffnen. Der gewählte modulare Ansatz erlaubt es, für einzelne Teilaufgaben jeweils besonders geeignete graphische Nutzerschnittstellen zu entwickeln und gemeinsam mit textuellen sowie visuellen Formen der Datendarstellung zu einer Gesamtschnittstelle zu kombinieren.
- Evolution
- ▶ Die Verwendung von miteinander kombinierbaren Nutzerschnittstellen für Anfragedefinition und Ergebnisauswertung unterstützt den erforderlichen iterativen Entwicklungsprozess der Schnittstelle in mehrfacher Hinsicht. Auf diese Weise können einzelne Nutzerschnittstellen jeweils unabhängig voneinander entwickelt und verbessert werden. Überdies kann die Zahl der bereitgestellten Module sukzessive erweitert werden, so dass die Funktionalität zur Datenselektion und zur Auswertung von Selektionsergebnissen iterativ über die Zeit ausgeweitet werden kann.

Es wurde erwartet, dass auf diese Weise den Datennutzern sowohl die Beherrschung der Schnittstelle, der Zugang zu einzelnen Datenräumen, die intuitive Formulierung individueller Selektionskriterien sowie die Auswertung von Ergebnisdaten erleichtert wird.

### 11.5 Flexibilität und Adaptivität

Um eine möglichst flexible Verwendbarkeit der Schnittstelle durch individuelle Datennutzer sicherzustellen, wurde Wert darauf gelegt, dass die zu entwickelnde Anwendungssoftware mit minimalem Aufwand an potentiell beliebigen Orten sowie auf unterschiedlichen Hard-

ware- und Betriebssystemplattformen eingesetzt werden kann. Ferner wurde die Schnittstelle so konzipiert, dass die Einbeziehung von Datenräumen nicht auf die Datenbanken des Institutes limitiert ist, sondern eine grundsätzliche Offenheit für potentiell beliebige relationale Datenbankmanagementsysteme an beliebigen Orten besteht.

Zusätzlich wurde ein hoher Grad von Adaptivität an die schrittweise Erweiterung der bereitzustellenden Datenschicht angestrebt. Hierfür wurden Konfigurationsmechanismen entwickelt, die eine unaufwendige Anpassung der Schnittstelle an einzelne Datenräume unterstützen und die Einbindung neuer sowie Änderungen an Strukturen bereits eingebundener Datenbanken *ohne Reprogrammierung*, d.h. ohne Eingriffe in den Programmcode der Software erlauben.