

TEIL E

ERREICHTER STAND, BEWERTUNG UND AUSBLICK

In diesem letzten Teil wird der im Rahmen der Arbeit erreichte Stand zusammengefasst und bewertet sowie ein Ausblick auf die für die nächsten drei Jahre geplante Ausweitung der Datenerschließung gegeben. Kapitel 23 fasst Aspekte zum Betrieb der Schnittstelle zusammen; hierunter fallen aktuelle Konfiguration, Größe des entstandenen Codes für den Client, die Bereitstellung von Clients für die Anwender sowie die Adaptierbarkeit der Schnittstelle. Kapitel 24 stellt Material zur Akzeptanz der Schnittstelle anhand des über sie zugänglichen Datenvolumens sowie der Zahl und Art der Zugriffe zusammen. In Kapitel 25 werden kurz einige Nachnutzungen vorgestellt, die von der Entwicklung der Schnittstelle geprägt wurden bzw. von ihren Komponenten profitiert haben; das abschließende Kapitel 26 bewertet den erreichten Stand und gibt einen Ausblick auf die darauf aufbauende, ab 2004 im Rahmen eines TOPIK-Projektes adressierte Erschließung weiterer Arten von Zeitreihen.

23 Betrieb

In diesem Kapitel werden Aspekte zum Betrieb der Schnittstelle zusammengestellt. Dies umfasst die aktuelle Konfiguration sowie die Größe der Clients (Kap. 23.1), die Bereitstellung von Clients auf den Anwenderrechnern (Kap. 23.2) sowie verschiedene Gesichtspunkte der Adaptierbarkeit der Schnittstelle (Kap. 23.3).

23.1 Aktuelle Konfiguration und Größe des Client

Die nachfolgende Tabelle (Tab. 23.1) gibt einen Überblick über die gegenwärtig im Potsdam-Institut für Klimafolgenforschung eingesetzte Konfiguration der Schnittstelle und die sich hieraus ergebende Verwendung der einzelnen Filtermodule und Auswertungsmodule für die eingebundenen Datenräume.

Name	Typ	PIK CERA-2	Zeitreihen- metadatenbanken			Zeitreihen- datenbanken		
			German	Global	DWD	German	Global	Scen.
SingleAttributeFilter	FM	✓	✓	✓	✓	✓	✓	✓
SpatialFilter	FM	✓	✓	✓	✓	✓	✓	✓
TimeFrameSelector	FM	✓	✓	✓	✓	✓	✓	✓
ValueCombinator	FM		✓	✓	✓	✓	✓	✓
GlobalSearchFilter	FM	✓						
HierarchyBrowser	FM	✓						
StationClassifier	FM		✓	✓	✓	✓	✓	✓
StatisticFilter	FM					✓	✓	✓
AttributeSelector	FM	✓ ³³¹	✓	✓	✓	✓	✓	✓
ResultNavigator	AM	✓						
EntryViewer	AM	✓						
BboxVisualizer	AM	✓						
TableView	AM		✓	✓	✓	✓	✓	✓
StationVisualizer	AM		✓	✓	✓	✓	✓	✓
PostProcessor	AM		✓	✓	✓	✓	✓	✓
TimeSeriesSelector	AM					✓	✓	✓
TimeSeriesVisualizer	AM					✓	✓	✓

Tab. 23.1 - Aktuelle Konfiguration der Schnittstelle, Stand Okt. 2003: Zuordnung der Filtermodule (Typ = FM) und Auswertungsmodule (Typ = AM) zu den einzelnen eingebundenen Datenräumen.

Zur Realisierung des adaptiven Client der Schnittstelle wurden insgesamt 152 verschiedene Java-Klassen entworfen und implementiert. Die daraus resultierende ebenso große Zahl von Bytecode-Dateien wird in einem Java Archive (JAR) komprimiert zusammengefasst und in dieser Form auf den Rechner des Anwenders geladen. In der gegenwärtig eingesetzten Version besitzt das resultierende JAR-File eine Größe von rund 410 Kilobyte (vgl. Tab. 23.2).

	Anzahl Klassen	Anzahl Quellcode-Zeilen		Größe des JAR-Files
		mit Kommentar ³³²	Lines of Code ³³³	
IDA	32	ca. 12.000	ca. 6.000	ca. 54 KB
Client mit IDA	152	ca. 41.000	ca. 23.000	ca. 410 KB

Tab. 23.2 - Dimensionen des entstandenen Client-Codes der Schnittstelle.

23.2 Bereitstellung des Client

In Betriebsphase I wurde der Client der Schnittstelle sowohl als Applet wie als Applikation (vgl. Kap. 13.3.2 und 13.3.3) betrieben. Eine Applet-Version konnte über Web-Browser durch Aufruf

³³¹ In PIK CERA-2 eingesetzt zum interaktiven Abruf von Informationen über die eingebundenen Attribute.

³³² Die Kommentierung des Quellcodes erfolgt direkt in den einzelnen Quellcode-Dateien.

³³³ Angegeben ist die Anzahl der unkommentierten Quellcodezeilen (NLOC, Noncommented Lines Of Code).

einer entsprechenden HTML-Seite gestartet werden; aus Performance-Gründen wurde für den Einsatz auf IBM-Workstations mit AIX³³⁴-Betriebssystem der Client zusätzlich als Applikation bereitgestellt und den entsprechenden Anwendern empfohlen. Für den komfortablen Aufruf der Applikation wurden von der Scientific Data Management Group Skripte entwickelt, die eine gegebenenfalls erforderliche Aktualisierung des Client transparent für den Anwender durchführen. Dies war für den Betrieb als Applet nicht erforderlich, allerdings mussten hier die durch die derzeit in den Browsern verfügbare Version der Virtuellen Maschinen - Version VM 1.0 - gegebenen Einschränkungen hingenommen werden³³⁵.

Aufgrund der Erfahrungen aus Betriebsphase I wird der Client in Betriebsphase II generell als Applikation bereitgestellt. Die Software wird dabei zentral vorgehalten und gewartet; über ebenfalls zentral bereitgestellte Skripte für unterschiedliche Hardware/Betriebssystem-Plattformen können Anwender über das Intranet des Institutes einen Client starten. Die Skripte übernehmen dabei transparent für den Anwender die erforderlichen Schritte. Es wird zunächst überprüft, ob die jeweils aktuelle Version der Client-Software auf dem Rechner des Anwenders vorliegt. Ist dies nicht der Fall, wird sie zunächst dorthin geladen; ansonsten wird die lokal vorhandene Version der Software verwendet. Nachfolgend erfolgt der Aufruf einer entsprechenden Virtuellen Maschine zum Start des Client. Der gewählte Mechanismus erwies sich als im praktischen Einsatz geeignet zur zentralen Wartung und komfortablen und schnellen institutsinternen Bereitstellung der Schnittstelle auf unterschiedlichen Anwenderrechnern; sowohl eine gegebenenfalls erforderliche automatische (und für den Anwender vollkommen transparente) Aktualisierung sowie der Start der Anwendung erfolgen in der Regel innerhalb einiger Sekunden.

Der Client führt seinerseits bei seinem Start eine Überprüfung der jeweils gültigen Versionsnummer durch, in dem er diese vom Server abfragt und mit seiner eigenen vergleicht. Sind beide Versionsnummern nicht identisch, wird der Anwender darauf hingewiesen, dass eine Aktualisierung erforderlich ist und ein Start des Client verweigert. Mit dieser Vorgehensweise werden zwei Ziele verfolgt. Zum einen wird so eine Verwendung der jeweils aktuellen Version auch für solche Anwender gewährleistet, die sich zwar im Intranet des PIK befinden, den Client jedoch nicht über eines der dafür vorgesehenen Skripte starten. Zum anderen kann auf diese Weise die Konsistenz von Clients, die institutextern bereitgestellt werden, sichergestellt werden. Solche externen Clients können in Absprache mit dem PIK zur Verfügung gestellt und bspw. über einen Web-Browser mittels HTTP-Download heruntergeladen werden. Diesbezügliche Erfahrungen zeigen, dass die hierfür erforderlichen Schritte unaufwendig erfolgen können; so nehmen Herunterladen und Bereitstellung für eine externe Erstbenutzung in der Regel nur wenige Minuten in Anspruch.

Grundsätzlich hängt die korrekte Ablauffähigkeit sowie die Performance eines Java-Programms von der jeweils eingesetzten Virtuellen Maschine zur Interpretation des Bytecodes ab. Insbesondere zu Beginn der Entwicklung waren je nach Kombination von Betriebssystem und Virtueller Maschine sporadisch kleinere Probleme mit den graphischen Oberflächen bei einzelnen Clients festzustellen, die heute kaum mehr auftreten. Eine uneinheitlicher Ablauf der internen Programmlogik der Schnittstelle auf unterschiedlichen Virtuellen Maschinen konnte hingegen *nicht* festgestellt werden. So wurde bspw. der Server problemlos auf unterschiedlichen Hardware- und

³³⁴ AIX ist die Unix-Variante von IBM.

³³⁵ So erforderten die limitierten Rechte beim Netzwerkzugriff, den Server der Schnittstelle auf dem selben Rechner zu installieren, auf dem auch der Web-Server, über den das Applet ausgeliefert wurde, betrieben wurde. Dies war zwar aufgrund der Realisierung des Servers als plattformunabhängige Anwendung problemlos möglich, stellt jedoch dennoch eine unerwünschte Limitierung dar. Zudem ergab sich die paradoxe Situation, dass Anfrageergebnisse, die von der Schnittstelle bereits auf den Rechner des Anwenders geladen wurden, vom Client-Applet zwar angezeigt, aber dort nicht direkt gespeichert werden konnten. Um dies dennoch zu ermöglichen, wurden Anfrageergebnisse, die der Anwender speichern wollte, vom Server erneut, diesmal über dynamisch generierte HTML-Seiten bereitgestellt. Diese wurden vom Applet im Browser des Anwenders geöffnet und konnten so lokal gespeichert werden.

Betriebssystemplattformen (IBM-Workstations mit unterschiedlichen AIX-Versionen; Intel-PC mit Windows 2000) eingesetzt; die Clients werden primär auf den Betriebssystemen AIX sowie verschiedenen Windows-Varianten betrieben, jedoch auch auf Linux sowie Apple-MacIntosh eingesetzt.

23.3 Adaptierbarkeit

Die Adaptierbarkeit der Schnittstelle bildete eine der zentralen Anforderungen. In diesem Kapitel werden verschiedene diesbezügliche Aspekte dargestellt.

23.3.1 Software

Die vorgesehene sukzessive Erweiterung der Software der Schnittstelle erwies sich aufgrund ihrer modularen und adaptiven Struktur als problemlos durchführbar. Erforderliche Fehlerbehebungen sowie Verbesserungen und Erweiterungen konnten dabei jeweils durchgeführt werden, ohne den laufenden Betrieb zu unterbrechen³³⁶. Client und Server der Schnittstelle liegen dazu jeweils in einer *production version*, die im laufenden Betrieb eingesetzt wird, und einer oder mehreren *working versions* vor, an denen erforderliche Änderungen durchgeführt und getestet werden. Ist eine *working version* betriebsbereit, wird sie zur neuen *production version* erklärt und zentral bereitgestellt; sie steht damit den Anwendern beim nächsten Aufruf eines Client automatisch zur Verfügung.

23.3.2 Einbindung neuer Kartendaten

Unproblematisch erfolgte auch die nachträgliche Erweiterung der über IDA bereitgestellten Kartenhierarchien. So wurden eine Hierarchie im laufenden Betrieb (Februar 2002) um Flusseinzugsgebietskarten für sämtliche Kontinente erweitert. Nachdem das entsprechende Kartenmaterial aus GIS-Daten generiert wurde (vgl. Kap. 17.3.5), genügte eine Änderung in der Konfigurationsdatei von IDA, um dieses einzubinden und den Anwendern bereitzustellen.

Die Adaption der Schnittstelle an neu hinzukommende oder sich verändernde Datenräume konnte ebenfalls unaufwendig und ohne Reprogrammierung durchgeführt werden. Dabei sind die Einbindung neuer Datenräume, die Erweiterung bereits eingebundener Datenräume sowie strukturelle Änderungen bereits eingebundener Datenräume zu unterscheiden.

23.3.3 Einbindung neuer Datenräume

Die Adaption der Schnittstelle an neu einzubindende Datenräume konnte im laufenden Betrieb jeweils unaufwendig durchgeführt werden. Die Einbindung konnte dabei im wesentlichen jeweils durch entsprechende Änderungen in der Konfigurationsdatei für den *Client* der Schnittstelle erfolgen. Hier können alle für die Adaption des Client erforderlichen Informationen festgelegt werden, bspw. welcher Datenbankgruppe der neue Datenraum zuzuordnen ist und welche Filtermodule und Auswertungsmodule dem Anwender zu dessen Erschließung zur Verfügung stehen sollen (vgl. Kap. 14.3). Sollen für eines oder mehrere der Attribute des einzubindenden Datenraumes zur Laufzeit Transformationen ihrer Werteausprägungen über den DVM-Mechanismus unterstützt werden (vgl. Kap. 16.2.2), sind zusätzlich die erforderlichen Look-Up-Tables bereitzustellen. Eine Rekonfigurierung des *Servers* ist nur erforderlich, wenn der neue Datenraum über ein relationales Datenbankmanagementsystem (RDBMS) bereitgestellt wird, auf das dieser noch keinen entsprechenden Zugriff besitzt. In diesem Fall ist die Konfigurationsdatei des Servers um die entsprechenden Informationen zu erweitern und gegebenenfalls ein weiterer JDBC-Treiber bereitzustellen; eine Reprogrammierung des Servers ist ebenfalls *nicht* erforderlich. Durch Bereitstellung der

³³⁶ So wurden bspw. das Auswertungsmodul BoundingBoxVisualizer (vgl. Kap. 21.1.5) sowie erweiterte Sortierfunktionen für den TableView (vgl. Kap. 21.2.3) nach dem Beginn von Betriebsphase II (Juni 2001) aufgrund von Anregungen von Anwendern im laufenden Betrieb entwickelt und nachträglich (Juli 2001) in die Schnittstelle eingefügt.

neuen Konfigurationsinformationen adaptieren sich die Clients automatisch bei ihrem nächsten Aufruf und stellen dem Anwender den neu eingebundenen Datenraum zur Verfügung.

23.3.4 Erweiterung eingebundener Datenräume

Da eine Erweiterung des Datenvolumens eines bereits eingebundenen Datenraumes durch zusätzlich in diesen integrierte Datensätze keine strukturellen Änderungen impliziert, war in solchen Fällen keine Rekonfiguration der Schnittstelle erforderlich. Werden in einen Datenraum hingegen solche Datensätze hinzugefügt, die bisher nicht vorhandene Werteausprägungen für Attribute enthalten, die über den DVM-Mechanismus transformiert werden sollen (vgl. Kap. 16.2.2), ist die Schnittstelle über eine entsprechende Erweiterung und Bereitstellung der Dateien mit den jeweiligen Look-Up-Tabellen zu adaptieren. Dies ist im praktischen Betrieb zum Beispiel der Fall, wenn neue Zeitreihenmetadaten mit bisher dort nicht dokumentierten Variablen zu einer bereits eingebundenen Datenbank hinzugefügt werden. Die Editierung der Look-Up-Tabellen erfordert minimale manuelle Eingriffe, die vor dem Hintergrund, dass die Bereitstellung neuer Variablen über die Schnittstelle stets auch mit einer vorherigen manuellen Erweiterung der entsprechenden Hilfedateien um diesbezügliche Informationen für den Anwender einhergeht, im bisherigen Betrieb nicht als Einschränkung empfunden wurden.

23.3.5 Strukturelle Änderung eingebundener Datenräume

Strukturelle Änderungen bereits eingebundener Datenräume ergaben sich beispielsweise durch nachträgliche Erweiterungen einzelner Datenbanken mit Zeitreihenmetadaten auf hierarchische räumliche Klassifikatoren für Flusseinzugsgebiete, für die ein entsprechendes Attribut (`Basin_ID`, vgl. Kap. 18.3.2) hinzugefügt wurde. Die Adaption der Schnittstelle an eine Veränderung der Struktur eines bereits eingebundenen Datenraumes durch Hinzufügen oder Entfernen von Attributen konnte im Bedarfsfall im laufenden Betrieb unaufwendig durchgeführt werden. Dies erfolgte jeweils durch entsprechende Änderungen der Konfiguration³³⁷ der Clients, die sich dadurch automatisch bei ihrem nächsten Aufruf an den veränderten Datenraum adaptieren.

³³⁷ Im Falle des Attributes `Basin_ID` waren hier die Filtermodule `SpatialFilter` (vgl. Kap. 20.3) sowie `AttributeSelector` (vgl. Kap. 20.10) entsprechend umzukonfigurieren, um sowohl eine entsprechende Raumauswahl bereitzustellen wie das neue Attribut für die Ergebnispräsentation zu aktivieren.