

Hochschule für Technik und Wirtschaft Dresden (FH)
Fachbereich Vermessungswesen / Kartographie

Diplomarbeit

**Aufbau einer GIS-Lösung zum Nachweis von Daten der
bathymetrischen Forschung am Alfred-Wegener-Institut**

Vorgelegt von: Sebastian Röber
Studiengang Vermessungswesen

Begutachtung:
Prof.Dr.-Ing. Frank Schwarzbach
Prof.Dr.-Ing. Gunter Reppchen

Betreuung am AWI:
Dr. Hans Werner Schenke
Dr. Martin Klenke

22. April 2004

Anmerkungen zur Lesbarkeit

Die Erstellung einer Internet/Intranet-Anwendung sowie der Einsatz der ArcGIS-Produkte der Firma ESRI bedingen in der vorliegenden Arbeit die häufige Verwendung von Begriffen aus dem englischen Sprachraum, da in den meisten Fällen eine Übersetzung ins Deutsche nicht zur Verständlichkeit beiträgt. Um die englischen Begriffe aber dennoch als solche zu kennzeichnen, müssen sie im laufenden Text hervorgehoben werden. Die folgenden Festlegungen sind dabei ein Kompromiss zwischen Gewohnheit, Lesefluss, inhaltlicher Richtigkeit und Verständlichkeit:

Server, Client¹ und Service sind zentrale Begriffe für webbasierte Anwendungen und werden im Text nicht hervorgehoben. Dagegen werden alle begrifflichen Eigennamen, wie z.B. *Shapefile*, *Geodatabase*, *Feature Class*, *Viewer*, *Application Server* ... durch Schrägstellung gekennzeichnet. Dies gilt aber nur, sofern es sich nicht um konkrete Software-Produkte handelt, wie z.B. der ArcIMS Application Server oder die ArcIMS Viewer.

Die bisher aufgeführten Begriffs-Typen werden aber insoweit dem Sprachgebrauch angepasst, dass ihnen, falls erforderlich, ein Genitiv- oder ein Mehrzahl-“s“ angehängt wird, wie z.B. die Clients oder des *Viewers*.

Des Weiteren werden alle Funktionsaufrufe, Befehle und Tabellennamen in einfache Anführungszeichen gesetzt, während alle im übertragenen Sinne gemeinten Begriffe von doppelten Anführungszeichen umschlossen werden, z.B. „Virtual Server“ oder „on the fly“.

Die vorliegende Arbeit wurde in doppelseitiger Ausführung angefertigt. Damit soll, neben der Lesbarkeit, vor allem ein Beitrag zum Umweltschutz geleistet werden, insbesondere vor dem Hintergrund, dass die Herstellung eines weißen A4 Blatts soviel Energie erfordert, wie nötig wäre, um eine 60-Watt-Glühlampe eine Stunde lang zu betreiben (Umweltbundesamt (1998), in „Umweltbewusst leben“, S. 335). Papier ist Energie!!

Sebastian Röber

¹Der Client ist rechnergestütztes System bestehend aus Hard- und Software, über das der Nutzer auf eine zentrale Anwendung zugreifen kann

Inhaltsverzeichnis

1	Einleitung	1
2	Die bathymetrische Datenverarbeitung am AWI	5
2.1	Datengewinnung und -aufbereitung	6
2.1.1	Positionsbestimmung	6
2.1.2	Tiefenmessung	7
2.1.3	Datenfluss/ Formate	9
2.2	Datenverwaltung	11
2.2.1	PODAS	11
2.2.2	PANGEA	11
2.2.3	Bathymetrische Datenverwaltung	13
2.3	Dateizugriff - Problemdarstellung	14
3	Konzeptionelle Anforderungen an ein neues System zur bathymetrischen Datenverwaltung	17
3.1	Zielsetzung - ein GIS?	17
3.2	Datentypen	19
3.2.1	Metadaten	19
3.2.2	Roh- und Ergebnisdaten	23
3.2.3	Projektdateien	24
3.3	Art der Datenablage	24
4	Unterstützende Software und Systeme	27
4.1	Windows/ Internet	27
4.2	Microsoft SQL Server 2000	28
4.3	ArcGIS-Produktfamilie	29
4.3.1	ArcGIS Desktop	30
4.3.2	ArcSDE	32
4.3.3	ArcIMS	35

5	Praktische Umsetzung	41
5.1	Konzeption des Datenmodells	42
5.2	Aufbau der Datenbank	45
5.3	Einlesen von Beispieldatensätzen in die Datenbank	48
5.3.1	Vorbereitung	48
5.3.2	Einlesen	50
5.4	Erstellen von standard-konformen Metadaten	53
5.5	Konzeption und Anlage einer ArcIMS Webseite	55
5.6	Projektorientierte Anpassung	61
6	Diskussion der Ergebnisse	65
6.1	Funktionalität/Handhabung	65
6.2	Ergebniseinschätzung	69
7	Zusammenfassung und Ausblick	71
	Literaturverzeichnis	73
	Danksagung	76
A	Anhang A	77
B	Quellcode	83
B.1	Programm 'NakKonverter'	84
B.2	Programm 'AsciiKonverter'	90
C	Metadaten	93
D	Javascript	99
E	Inhalt der beigefügten CD	105

Abbildungsverzeichnis

1.1	Forschungsschiff Polarstern	2
2.1	Darstellung einer Fächersonarmessung	8
2.2	PODAS-Datenbank	12
2.3	PANGEA-Datenbank	12
3.1	Catalog Service	20
4.1	ArcGIS-Produktfamilie	30
4.2	ArcSDE (Datenbankgateway)	33
4.3	Räumlicher Index	34
4.4	Architektur des ArcIMS	36
4.5	Antwort des Spatial Servers	37
4.6	Aufbau des ArcIMS	38
5.1	Datenmodell	44
5.2	Einrichten einer räumlichen Datenbankverbindung	46
5.3	Anlage eines <i>Feature Dataset</i>	47
5.4	Anlage einer <i>Feature Class</i>	47
5.5	NAK-Datei	49
5.6	Aufruf des Programms 'NakKonverter'	50
5.7	Linien-Coverage	51
5.8	Funktion: 'Create Feature Class -> From XY Table'	52
5.9	FGDC Metadateneditor	54
5.10	ArcIMS Author	56
5.11	<i>Map Configuration File</i>	57
5.12	Einrichten der ArcIMS Services	57
5.13	Java Custom-Viewer	60
5.14	Aufruf der Javascript-Funktion 'showSelect()'	61
5.15	Zuweisen der Internet-Links	62
5.16	Abfrage der Ergebnisdatensätze	63

Abbildungsverzeichnis

6.1	Analyse-Funktionen	66
6.2	Aufruf der 'Attributes'-Funktion	66
6.3	Der 'Query Builder'	67
6.4	Projektbezogene Analyse-Funktionen	68
6.5	Hinzufügen von Daten-Layern	68
A.1	Die <i>Geodatabase</i>	78
A.2	<i>Personal/ArcSDE Geodatabase</i>	79
A.3	Von ArcIMS unterstützte Bildformate	80
A.4	Funktionsumfang der ArcIMS Services	81

Tabellenverzeichnis

5.1	Abschätzung des Datenvolumens der Tiefenmesssysteme für einen Tag . . .	43
-----	---	----

Abkürzungsverzeichnis

AWI	Alfred Wegener Institut in Bremerhaven
Beams	Strahlenbündel
CSDGM	Content Standard for Digital Geospatial Metadata
DBMS	Datenbank-Managementsystem
DGM	Digitales Geländemodell
DGPS	Differential Global Positioning System
DOLOG und EM-Log	zwei nautische Systeme
DWS	Deep Water Sounder
ESRI	Environmental System Research Institute
FGDC	Federal Geographic Data Committee
GEODAS	Geophysical Data System
GMT	Generic Mapping Tool
GPS	Global Positioning System
HDBE	High Definition Bearing Estimation
HDCS	Hydrographic Data Cleaning System
HIPS	Hydrographic Information Processing System
INMARSAT	International Maritime Satellite
MINS	Marine Inertial Navigation System
NACOS	Integrated Navigation and Command System
NSDI	National Spatial Data Infrastructure
PANGEA	PaleoNetwork for Geological and Environmental Data
PODAS	Polarstern Data System
Polarstern	Forschungsschiff „Polarstern“ des AWI
RDBMS	Relationales Datenbank-Managementsystem
SURF	Sensor-Unabhängiges-Rohdaten-Format
SQL	Standard Query Language
XML	Extended Markup Language

1 Einleitung

Die Vermessung der Meeresbodentopographie ist wesentliche Grundlage für viele wissenschaftlichen Disziplinen der Meeresforschung. In den marinen Bereichen der Geologie, Geophysik sowie der Biologie und Ozeanographie dienen digitale Geländemodelle (DGM) zur Interpretation und zur räumlichen Zuordnung der jeweiligen Messdaten. Aber auch in den technischen Bereichen werden DGMs verwendet, wie z.B. zur Herstellung von Seekarten oder als Planungsgrundlage für den Meeresbergbau (Schenke (1990)).

Die Arbeitsgruppe Bathymetrie des Alfred-Wegener-Institutes (AWI) in Bremerhaven befaßt sich mit der Vermessung, Beschreibung und Darstellung der Meeresbodentopographie. Die Hauptuntersuchungsgebiete liegen dabei jeweils in den polaren Breiten (s. Abb. 1.1). Auf Grundlage systematischer Vermessungen mit Fächersonarsystemen, die eine flächenhafte Erfassung des betreffenden Untersuchungsgebietes ermöglichen, werden digitale Geländemodelle (DGM) berechnet. Diese dienen als Ausgangspunkt, um weitere kartographische Darstellungsformen (z.B. bathymetrische Karten) abzuleiten.

Allerdings ist entwicklungsbedingt der für diese Produkte zugrundeliegende bathymetrische Datenbestand hinsichtlich Qualität, Prozessierungsstand, Datenformat und Speichermedium äußerst heterogen. Durch das dateibasierte Ablagesystem ist zudem eine feste Verknüpfung der Datensätze mit Metadaten nicht gegeben, was zu zusätzlichen Redundanzen in der Datenhaltung führt. Datenrecherche und -analyse sowie Datenextraktion werden aufgrund dieser Tatsachen erheblich erschwert. Da das Datenaufkommen pro Zeiteinheit durch die Weiterentwicklung der Messtechnik in Zukunft noch steigen wird, sind die konzeptionellen Grenzen des aktuellen Datenverwaltungssystems bald erreicht.

Im Rahmen der vorliegenden Arbeit soll deshalb ein geographisches Informationssystem aufgebaut werden, das die Verwaltung des bathymetrischen Datenbestandes erlaubt und die Datenrecherchen und -analysen erleichtert. Dazu ist es zum einen notwendig, ein Datenmodell zu erstellen, in dem alle vorhandenen Datentypen integriert und miteinander verknüpft werden können und zum anderen eine graphische Oberfläche zu schaffen, über die der Datenbestand abfragbar ist. Um auch Nutzern außerhalb der Arbeitsgruppe der Bathymetrie die Möglichkeit zu geben, mittels Internet/Intranet-Anwendungen auf den



Abbildung 1.1: Forschungsschiff Polarstern auf Expedition

bathymetrischen Datenbestand zuzugreifen, ist der Einsatz eines Internet Map Servers erforderlich. Auf der Grundlage seiner Funktionalitäten und Eigenschaften wird das Datenmodell konzipiert.

Bei der Umsetzung des Datenmodells in ein geographisches Informationssystem (GIS) werden die am AWI vorhandenen Softwarekomponenten genutzt und gegebenenfalls angepasst. Im Mittelpunkt der Anwendung stehen dabei vor allem die ArcGIS-Produkte der Firma ESRI (Environmental System Research Institute). Um die Funktionsweise und die Handhabung des GIS veranschaulichen zu können, wurden Beispieldatensätze entsprechend aufbereitet, in das System importiert und mit Metadaten verknüpft.

Das Ziel der vorliegenden Arbeit ist es, eine graphische Oberfläche zu erstellen, über die der Nutzer in der Lage ist, auf den kompletten bathymetrischen Datenbestand zuzugreifen.

Im Folgenden wird zunächst in Kapitel 2 eine kurze Einführung in die bathymetrische Datengewinnung und -aufbereitung gegeben, bevor die Datenverwaltung sowie die sich daraus ergebenden Probleme näher betrachtet werden. Danach folgt in Kapitel 3 die Formulierung der Zielsetzung, sowie die Klassifizierung der vorhandenen Daten. Außerdem werden mit der Datenorganisation erste methodische Grundlagen für die Konzeption des Datenmodells gelegt. In Kapitel 4 werden die verwendeten Software-Komponenten vorgestellt. Dabei werden jeweils nur die Funktionen ausführlicher beschrieben, die in der praktischen Umsetzung

auch zum Einsatz kommen. Die Konzeption des Datenmodells, der Aufbau des GIS sowie der Import von Beispieldatensätzen werden in Kapitel 5 erläutert. Außerdem wird sowohl die Erstellung von standard-konformen Metadaten als auch die Anlage der Webseite beschrieben. In Kapitel 6 erfolgt die Ergebnisdiskussion und in Kapitel 7 wird die Arbeit noch einmal zusammengefasst.

2 Die bathymetrische Datenverarbeitung am AWI

Die Kenntnis der Topographie des Meeresbodens ist für die Interpretation von chemischen, physikalischen, biologischen und geologischen Prozessen im Bereich der Meeresforschung unerlässlich. Um die topographischen Gegebenheiten des Meeresbodens abzubilden, werden von der Arbeitsgruppe Bathymetrie auf Grundlage von Echolotmessungen digitale Geländemodelle berechnet und bathymetrische Karten erstellt.

Die Aussagefähigkeit der digitalen Geländemodelle und der daraus abgeleiteten Karten wird durch die Qualität und Quantität der zugrundeliegenden bathymetrischen Daten bestimmt. Eine entsprechend hohe Datendichte innerhalb eines bestimmten Gebietes lässt sich zum einen durch den Einsatz von geeigneten Messinstrumenten (z.B. Fächersonare) und zum anderen durch die Kombination der Messungen verschiedener Expeditionen erreichen. Dabei ist es gerade für Letzteres wichtig, die aufgezeichneten Daten so zu verwalten und zu organisieren, dass sie anhand ihrer geographischen Lage und ihrer inhaltlichen Eigenschaften wiedergefunden und verwendet werden können. Angaben zur Qualität der Daten können nur mit Kenntnis der jeweiligen Messanordnung, der eingesetzten Messmittel und der angewendeten Auswerteverfahren gemacht werden.

Im Folgenden soll daher zunächst eine Einführung¹ in die aktuellen Verfahren zur Positionsbestimmung und zur Tiefenmessung sowie in die Datenauswertung gegeben werden. Der zweite Teil des Kapitels beschäftigt sich mit der Datenverwaltung am AWI und in der Arbeitsgruppe Bathymetrie. Im Vordergrund stehen dabei vor allem die Architektur der Datenablage und die Möglichkeiten des Datenzugriffs. Im letzten Teil werden die Probleme dargestellt, die beim Arbeiten mit bathymetrischen Daten aufgrund des aktuellen Verwaltungssystems auftreten.

¹Die detaillierte Beschreibung der einzelnen Verfahren kann den angegebenen Literaturquellen entnommen werden.

2.1 Datengewinnung und -aufbereitung

2.1.1 Positionsbestimmung

Die Positionsbestimmung auf der Polarstern erfolgt über die Kombination verschiedener Messsysteme². Dabei sind die zwei inertialen Navigationssysteme (MINS)³, sogenannte Ring-Laser-Kreisel der Firma Raytheon-Anschütz⁴, die zwei GPS-Empfänger von Trimble, welche jeweils einem MINS fest zugeordnet sind, der Leica-GPS-Empfänger sowie die nautischen Systeme DOLOG und EM-Log⁵ von besonderer Bedeutung.

Jedes inertielle Navigationssystem verarbeitet neben den intern registrierten Beschleunigungswerten die Signale des zugehörigen Trimble-Empfängers und des EM-Logs des Schiffes. Daraus werden unter anderem Kurs-, Roll- und Stampfwinkel sowie Positionsdaten ermittelt und zum Einen an die Hydrosweep-Anlage, das Parasound sowie den Deep Water Sounder (s. 2.1.2 Tiefenmessung) und zum Anderen an das integrierte Navigationssystem (NACOS)⁶ weitergegeben (Gerchow 2001). Die dabei erreichte Positionsgenauigkeit beträgt durchschnittlich 10 Meter. Es ist stets nur eines der beiden verfügbaren Systeme „Online“, weshalb nur jeweils eine Datenübertragung stattfinden kann. Das andere MINS fungiert währenddessen als Backup-System.

Im Gegensatz zu der reinen GPS-Absolutpositionierung der Trimble-Empfänger bezieht der Leica-GPS-Empfänger über eine INMARSAT⁷-Satellitenverbindung Korrekturwerte der Referenzstation zur Ermittlung einer DGPS-Position. Diese Position findet ebenfalls Eingang in das NACOS.

Das NACOS bestimmt, durch Kombination der Positionen und unter Einbeziehung der nautischen Systeme mittels eines integrierten mathematischen Koppelfilters, die sogenannte „System-Position“, die in einem Intervall von einer Sekunde an das Polarstern Data System (s. 2.2.1 PODAS) übertragen wird⁸ (Greitemann-Hackl 2001, S. 1 ff). Die Genauigkeit dieser Position beträgt auf offener See ebenfalls etwa 10 Meter, kann aber in Küstennähe (Entfernung der Referenzstation kleiner 2000 km), aufgrund der stabileren Korrektur-

²Beschreibung der derzeitig installierten Anlage; Informationen zu den vorherigen Systemen s. Schenke (1990) und Christen (1999)

³MINS - Marine Inertial Navigation System

⁴vgl. <http://www.raytheon-marine.de/highseas/pdf/brochures/mins.pdf>; Datum des Zugriffs: 09.02.2004

⁵Doppler-Log und elektromagnetisches Log

⁶NACOS - Integrated Navigation and Command System der ATLAS Hydrographics GmbH

⁷INMARSAT - International Maritime Satellite

⁸Ursprünglich wurde diese Position ebenfalls, statt der MINS-Position, an das Hydrosweep und das Parasound übertragen. Allerdings ist durch die Anwendung des Koppelfilters nicht mehr nachvollziehbar, welche Eingangsgrößen verwendet wurden. Man entschied sich daher für eine stetige Positionslösung für die Tiefenmessgeräte. Dem DWS wird erst in PODAS über den Zeitstempel eine Position zugewiesen.

datenübertragung für die DGPS-Position des Leica-Empfängers, wesentlich präziser sein (Rottmann et al. 1994).

2.1.2 Tiefenmessung

Das Grundprinzip einer Tiefenmessung mit dem Echolot besteht in der Messung der Laufzeit, die ein Schallimpuls benötigt, um die Wassersäule zu durchlaufen. Dazu wird von der Sendeeinheit des an der Schiffsunterseite angebrachten Schwingers, ein Schallimpuls vertikal (lotrecht) ausgesendet und nach Reflexion am und im Meeresboden von der Empfängerinheit des Schwingers wieder aufgenommen. Mit der Laufzeit des Schallsignales und einer bekannten mittleren Wassergeschwindigkeit lässt sich die Wassertiefe, bezogen auf das Niveau der Schwingereinrichtung, berechnen.

Bis auf den Navigation Sounder erhalten alle Sonarsysteme neben den Positionsdaten auch die Korrekturwerte für die Hub-, Roll- und Stampfbewegungen des Schiffes von dem MINS. Aufgrund dieser Verbesserungen wird die vertikale Aussendung des jeweiligen Schallimpulses erreicht.

Navigation Sounder Der LAZ 4420 von ELAC Nautik ist ein Einstrahlecholot. Er dient vor allem der Navigation im Flachwasser, das heißt mit seiner Hilfe wird verhindert, dass die Polarstern in unbekanntem Gewässern oder in Hafeneinfahrten auf Grund läuft. Die ermittelten Tiefenwerte werden nicht für bathymetrische Zwecke verwendet.

Deep Water Sounder (DWS) Der EA 500DWS der Firma SIMRAD ist ebenfalls ein Einstrahlecholot, wird jedoch auf der Polarstern vorwiegend zur Bestimmung großer Tiefen eingesetzt. Aufgrund des kleineren Öffnungswinkels der Sendeeinheit und der damit verbundenen kürzeren Laufzeit des Schallsignals, ist ein höherer Messimpuls möglich als beim Hydrosweep. Dieses Echolot erfordert ebenso wie das Navigationslot keine zusätzliche Bedienung durch einen Operator. Somit wird in der Tiefsee eine kontinuierliche Datenerfassung auf jeder Polarstern-Fahrt sichergestellt.

Hydrosweep DS2 Das Hydrosweep DS2 (Hydrographic multibeam sweeping survey echosounder) von der ATLAS Hydrographics GmbH ist im Gegensatz zu den vorher beschriebenen Systemen ein Fächersonarsystem. Statt des einen stark vertikal gebündelten Strahls wird hier nun rechtwinklig zum Schiffsweg ein breiter Schallimpuls abgestrahlt (s. Abb. 2.1⁹). Durch diesen einen Impuls (Ping) entsteht ein ganzer Fächer, der eine kontinuierliche und

⁹Quelle: <http://www.awi-bremerhaven.de/GEO/Bathymetry/allg/worktasks.html>; Datum des Zugriffs: 15.04.2004

flächenhafte Erfassung des Meeresbodens ermöglicht. Die Auffächerung des Pings erfolgt erst beim Empfang, indem die vom Meeresboden reflektierten Signale entsprechend ihrer physikalischen Charakteristik zu einzelnen Beams (Strahlenbündel) zusammengefasst werden (Christen 1999, S. 19 ff). Aus jedem Ping werden aufgrund dieser Signalverarbeitung an der Empfangseinheit 59 Beams quer zur Schiffslängsachse erzeugt, d.h. 59 einzelne Laufzeiten bestimmt. Daraus können dann Tiefen abgeleitet werden. Außerdem lassen sich aus der von der Empfangseinheit aufgenommenen Rückstreuung (engl. „backscatter“) und durch die Messung von Side Scan Sonar Daten Rückschlüsse auf die Meeresbodenbeschaffenheit ziehen.

Durch die sogenannte Querkalibrierung ist es diesem Fächersonarsystem möglich, „wahre“ Tiefen, bezogen auf eine gemessene Wasserschallgeschwindigkeit, zu bestimmen (Schenke 1992,). Die Datengenerierung und -aufzeichnung bei Hydrosweep und Parasound erfordern ein hohes Maß an Bedienung und Überwachung, weshalb nicht automatisch jede Polarsternfahrt Daten dieser Systeme liefert.



Abbildung 2.1: Darstellung einer Fächersonarmessung

Parasound Das ebenfalls von der ATLAS Hydrographics GmbH entwickelte Sedimentecholot Parasound (Parametric Echosounder) ist wie der DWS ein Einstrahlecholot. Allerdings wurde es nicht konstruiert, um präzise Tiefen zu ermitteln, sondern es soll vielmehr Auskunft über die Meeresbodenbeschaffenheit geben. Aufgrund des sehr kleinen Öffnungswinkels und der niedrigen Impulsfrequenz ist dieses System in der Lage, je nach Bodenverhältnissen, bis zu 200 Meter in den Meeresboden einzudringen (Frahm 2003, S. 20 ff). Damit können verschiedene Sedimentschichten erkannt und ihr Aufbau punktuell analysiert werden. Ebenso wie beim DWS sind die Tiefen beim Parasound nicht auf eine gemessene,

sondern auf eine mittlere Wasserschallgeschwindigkeit von 1500m/s normiert. Somit ist keine direkte Vergleichsmöglichkeit mit den Hydrosweep-Tiefen gegeben.

2.1.3 Datenfluss/ Formate

Der folgende Abschnitt beschäftigt sich mit der Auswertung bathymetrischer Daten. Er gibt einen Überblick zu den Bearbeitungsschritten, die aufgrund der Messanordnung, der Messverfahren und der auftretenden Störeinflüsse notwendig werden. Die sich jeweils daraus ergebenden neuen Dateien bilden die Grundlage für die weitere Diskussion.

Hydrosweep Auf der Polarstern werden während der Messung im Online-System HYDROMAP ONLINE zunächst alle durch das Fächersonarsystem gewonnenen Informationen in einem binären Rohdatenformat (ROH) auf Tape und Festplatte gespeichert. Für eine weitere Bearbeitung der Daten erfolgt eine Konvertierung in das SURF-Format¹⁰. In diesem Format sind neben den Messwerten (Laufzeiten, Rückstreuwerte) auch sämtliche Parameter (Neigungswinkel, Geschwindigkeitsangaben) abgelegt, welche für das korrekte Betreiben der Hydrosweep-Anlage notwendig sind. Im SURF erfolgt dann im Onlinesystem die Navigationseditierung. Diese wird wegen des Auftretens von Positionslücken und -sprüngen in den Navigationsdaten erforderlich. Ursachen dafür können in dem Einfluss der Änderung der Satellitenkonstellation, in der Abschattung durch Schiffsaufbauten oder in anderen Störeinflüssen (Mehrwegausbreitungen, Witterung) liegen¹¹ (s. 2.1.1 Positionsbestimmung). Durch das Löschen von extremen Ausreißern und das Zwischeninterpolieren einer oder mehrerer Positionen, versucht man einen, der Wirklichkeit entsprechenden, stetigen Verlauf des Schiffskurses zu rekonstruieren (Christen 1999, S. 44 f).

Aufgrund unterschiedlicher Einflüsse bei der Bestimmung der Laufzeit des Schallimpulses (Wasserschallgeschwindigkeit, Rückstreueigenschaften des Meeresbodens, Schiffseigengeräusche) ist neben der Navigations- auch eine Tiefeneditierung notwendig. Durch die Überlagerung der Tiefen aus sich überlappenden benachbarten Messprofilen lassen sich zum einen durch zufällige Fehler entstandene Ausreißer und zum anderen durch Systematiken verfälschte Messungen ermitteln und eliminieren. Dabei werden diese Daten nicht endgültig gelöscht, sondern als nicht weiter zu verarbeitende markiert („flagging“). Damit sind vorgenommene Änderungen jederzeit rücksetzbar.

¹⁰Das SURF-Format (Sensor-Unabhängiges-Rohdaten-Format) wurde von ATLAS Hydrographics als ein standardisiertes, plattformunabhängiges Datenformat für deren Fächersonarsysteme entwickelt.

¹¹Vor allem in der Vorgängerkonfiguration des jetzigen Navigationssystem machten sich diese Einflüsse besonders bemerkbar, da die Positionierung über eine DGPS-Lösung erfolgte und man so auf eine stabile INMARSAT-Satellitenverbindung für die Übertragung der Korrekturdaten angewiesen war (vgl. Christen (1999)).

Bei der Tiefeneditierung muss zwischen den beiden in der Arbeitsgruppe Bathymetrie vorhandenen Computerplattformen UNIX und Windows unterschieden werden. Findet sie unter UNIX mit der Software HDCS¹² statt, so ist zuvor eine weitere Datenkonvertierung in das DUX¹³-Format notwendig. Nach der erforderlichen Bearbeitung im HDCS werden diese Daten in das AWI-interne binäre HYD-Format (Hydrosweep) konvertiert, das neben der jeweiligen Zeitzuordnung nur noch die Koordinaten mit den zugehörigen Tiefenwerten enthält. Dieses bildet die Grundlage auf der einen Seite für die Ausgabe von Koordinaten-tripeln im ASCII-Format zur DGM-Berechnung und auf der anderen Seite zur Generierung von NAK¹⁴-Dateien, die u.a. für jede ermittelte Tiefe einen Zeitstempel und eine Position enthalten (s. 5.3 Einlesen von Beispieldatensätzen in die Datenbank). Die NAK-Dateien werden vor allem wegen der korrigierten Navigation auch von anderen Arbeitsgruppen (Geologie, Geophysik) des AWIs genutzt.

Unter Windows erfolgt die Tiefeneditierung im Programm CARIS HIPS (die PC-Version von HDCS) direkt im SURF. Allerdings ist ein Konvertieren in das HYD-Format dort nicht möglich, so dass der Weg wieder über UNIX führt. Den Endpunkt der Datenbereinigung bilden somit das SURF oder das DUX-Format, die danach entsprechend auf Datenbändern archiviert werden¹⁵.

Die Erzeugung des DGMs und die Berechnung von Isolinien in ArcINFO (ESRI) sind neben der kartographischen Gestaltung des Layouts die letzten notwendigen Schritte zur Erstellung bathymetrischer Karten. Diese Karten stellen somit das Endprodukt der beschriebenen Auswertung und ein wesentliches Ziel der bathymetrischen Forschungen dar.

Parasound Mit Hilfe des Datenerfassungsprogramms ParaDigMA werden auf der Polars-tern alle gemessenen Parasound Seismogramme im binären PS3-Format gespeichert. Alle weiteren notwendigen Bearbeitungsschritte erfolgen in diesem Format im Programm SeNT. Ebenso wie bei der Hydrosweep-Auswertung ist auch hier eine Navigationseditierung notwendig. Diese kann über verschiedene Filteralgorithmen und Interpolationsansätze erfolgen oder indem die vorhandenen Navigationsdaten über einen zeitlichen Vergleich durch die positionskorrigierten Daten des Hydrosweep-Systems, die sich in den NAK-Dateien befinden, ersetzt werden.

Da die weitere Auswertung eher von geologischem und geophysikalischem Interesse ist, sei an dieser Stelle nur auf weiterführende Literatur wie Iffland (2004) und Spieß (1993) verwiesen.

¹²HDCS - Hydrographic Data Cleaning System - ist Bestandteil des komplexen Programmsystems HIPS (Hydrographic Information Processing System)

¹³Datenformat der Firma ATLAS; nach dem Entwickler benannt

¹⁴Navigations-Korrigiert, internes AWI-Datenformat

¹⁵Es wird derzeit an entsprechenden Konvertieren gearbeitet, damit das SURF in Zukunft als einheitliches Bearbeitungs- und Archivierungsformat genutzt werden kann.

2.2 Datenverwaltung

2.2.1 PODAS

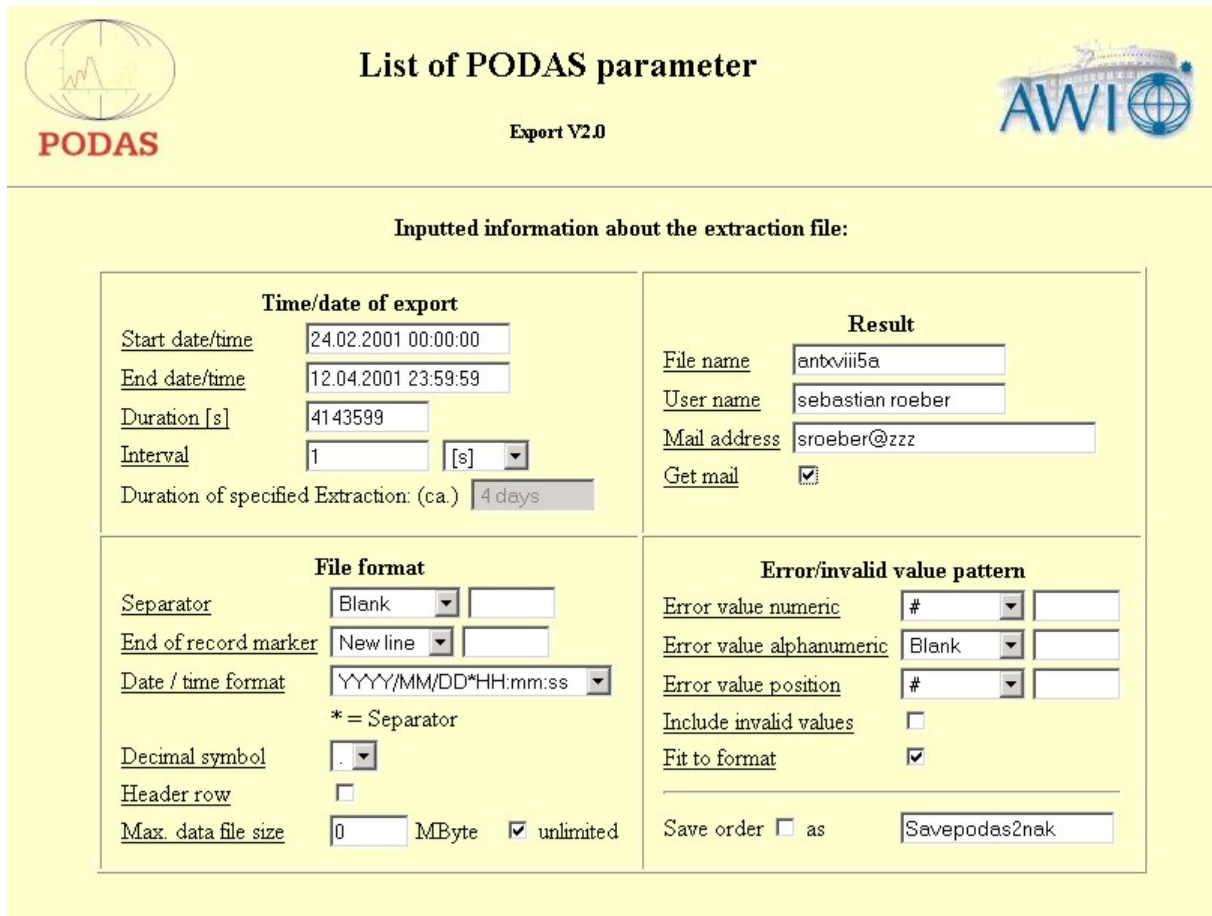
Das PODAS (Polarstern Data System) ist eine Datenbank, die auf der Polarstern systematisch nautische und wissenschaftliche Parameter von schiffsseitig installierten, kontinuierlich betriebenen Messeinrichtungen (Meteorologie, Ozeanographie, MINS), die der Beschreibung des aktuellen Zustandes dienen, erfasst. Diese Daten werden aufbereitet, gespeichert und den Nutzern für Selektionen und Auswertungen bereitgestellt (Werum 2001). Die Parameter von Sensoren und Messgeräten dagegen, die nur für bestimmte Projekte zum Einsatz kommen, werden nicht, oder nur zu einem sehr geringen Teil, in PODAS abgelegt, sondern von den jeweiligen Systemen selbst gespeichert. Dazu gehören unter anderem Hydrosweep, Parasound und Gravimetrie.

Die Datenextraktion im PODAS erfolgt dialoggesteuert über einen Webbrowser (s. Abb. 2.2). Die Zusammenstellung der gewünschten Daten (Sensorik, Zeitbereich, Datenauflösung) ist dabei durch jeden Nutzer individuell möglich, indem die entsprechenden Optionen in den einzelnen Dialogen gewählt werden. Außerdem können eine Vielzahl von Einstellungen für das Ausgabeformat vorgenommen werden. Die eigentliche Datenextraktion erfolgt dabei im Hintergrund, also „Offline“, durch ein Serverprogramm von PODAS. Nach der Durchführung des Auftrages wird der Nutzer per E-Mail informiert. Alle Ergebnisse werden in ASCII-Dateien auf einem FTP-Server für den Anwender zugänglich abgelegt. Die Ergebnisdateien sind dabei tabellenartig aufgebaut, d.h. je Zeile werden immer alle Werte der selektierten Sensoren zu dem jeweiligen Zeitpunkt aufgelistet. Somit ist eine einfache Weiterverarbeitung unmittelbar möglich (s. 5.3 Einlesen von Beispieldatensätzen in die Datenbank).

2.2.2 PANGEA

PANGEA (PaleoNetwork for Geological and Environmental Data) ist ein Informationssystem zur Verarbeitung, Archivierung und Veröffentlichung von georeferenzierten Daten der naturwissenschaftlichen Grundlagenforschung am AWI. Es verwaltet im Gegensatz zu PODAS vor allem projektbezogene Daten von der Polarstern. Das zugrundeliegende Datenmodell ist daher sehr an die Struktur des Messprozesses angepaßt: Verschiedene Projekte/Institute (PROJECT) sind an einem Forschungsbereich beteiligt, indem verschiedene Expeditionen (CAMPAIGN) durchgeführt werden, um an bestimmten geographischen Lokationen (EVENT) Meßreihen (DATA) aufzunehmen¹⁶. Diese einzelnen Hauptebenen werden mit Metadaten entsprechend ergänzt. Durch dieses einfache und flexible Datenmodell

¹⁶Eine vollständige Darstellung des Datenmodells kann Diepenbroek et al. 1999 entnommen werden.




List of PODAS parameter
Export V2.0

Inpupped information about the extraction file:

Time/date of export	Result
Start date/time: 24.02.2001 00:00:00	File name: antxviii5a
End date/time: 12.04.2001 23:59:59	User name: sebastian roeber
Duration [s]: 4143599	Mail address: sroeber@zzz
Interval: 1 [s]	Get mail: <input checked="" type="checkbox"/>
Duration of specified Extraction: (ca.) 4 days	

File format	Error/invalid value pattern
Separator: Blank	Error value numeric: #
End of record marker: New line	Error value alphanumeric: Blank
Date / time format: YYYY/MM/DD*HH:mm:ss	Error value position: #
* = Separator	Include invalid values: <input type="checkbox"/>
Decimal symbol: .	Fit to format: <input checked="" type="checkbox"/>
Header row: <input type="checkbox"/>	Save order <input type="checkbox"/> as: Savepodas2nak
Max. data file size: 0 MByte <input checked="" type="checkbox"/> unlimited	

Abbildung 2.2: Dialogfenster für die Eingabe von Parametern für die Datenextraktion aus der PODAS-Datenbank



PANGAEA

Data Software Info Links

PangaVista | ART | Projects | Institutes | PanCore

Search for: antxviii5a

-50 -180 180 -90

info@pangaea.de

Abbildung 2.3: Eingabemaske für eine Datenbankrecherche in PANGAEA

gelingt es, eine Vielzahl von heterogenen und dynamischen Daten zu integrieren.

PANGEA basiert auf einer 3-Ebenen (engl.: „three tiered“) Client/ Server Architektur. Auf der Client Seite (frontend-first tier) stehen eine Anzahl von Oberflächen (PangaVista, ART, PanCore, 4D Client) zur Verfügung, die die Analyse des vorhandenen Datenbestandes ermöglichen. Außerdem wird mit diesen Tools über entsprechende Zugriffsrechte auch der Import (4D Client) und Export von Daten in bzw. aus der Datenbank realisiert. Die Middleware (second tier) ist ein *Application Server*¹⁷, der verschiedene Funktionalitäten, wie Import und Suchfunktionen, softwareseitig realisiert (s. 4.3.3 ArcIMS). Die dritte Ebene (backend-third tier) ist das relationale Datenbank-Managementsystem SYBASE, in dem alle Daten abgelegt und verwaltet werden.

Außerhalb dieser Client/ Server Architektur stehen noch weitere Werkzeuge (PanMap, PanPlot, ODV) zur Verfügung, die der Visualisierung und Analyse herunter geladener Daten dienen (Diepenbroek et al. 1999 und Grobe et al. 1997). Durch diese zahlreichen zusätzlichen Programme wird zwar eine vielfältige und flexible Anwendung von PANGEA durch verschiedene wissenschaftliche Teildisziplinen erreicht, aber es wird gleichzeitig der Zugang für „unerfahrene Nutzer“ erschwert, da zum Einen eine alles verknüpfende intuitive Oberfläche fehlt und zum Anderen immer erst nach der abgeschlossenen Suche eine Visualisierung der Ergebnisse möglich ist. Der neue PANGEA Mapserver¹⁸ zur Abfrage u.a. von Parasound-Daten, beschreitet dahingehend genau den umgekehrten Weg, d.h auf der Grundlage von graphisch Dargestelltem können entsprechende Analysefunktionen aufgerufen werden. Dies erleichtert erheblich das Selektieren und Abfragen von Daten.

2.2.3 Bathymetrische Datenverwaltung

In der Arbeitsgruppe der Bathymetrie gibt es zur Zeit noch kein einheitliches System, z.B. in Form einer Datenbank, um bathymetrische Daten zu verwalten. Alle bearbeiteten Daten wurden (und werden) daher in nutzerspezifischen Verzeichnisstrukturen organisiert und dateibasiert gespeichert. Dazu standen entwicklungsbedingt verschiedene Speichermedien zur Verfügung. Kamen in der Anfangszeit von der Polarstern (ab 1982) in der Bathymetrie noch Magnetbänder (9-Spur-Streamer) mit einem Datenvolumen von 30 Megabyte zum Einsatz, so waren es schon Ende der 80er Jahre die TK 50/TK 70 Bänder mit einer Speicherkapazität von 90 bzw. 300 Megabyte. Diese wurden dann in den 90er Jahren von DAT- (Digital Audio Tape, 2GB) und DLT- (Digital Linear Type, 35GB) Bändern abgelöst. Trotz einer verbesserten Verwaltungsstruktur¹⁹ sind die Datenbänder hinsichtlich Zugriffsgeschwindig-

¹⁷Webserver-Software als Bindeglied zwischen Server und Client

¹⁸befindet sich derzeit in der Entwicklung

¹⁹Aufgrund des Einfügens von Verzeichnis- und Verwaltungsblöcken beim Beschreiben des Bandes, muss, um eine bestimmte Datei zu finden, nicht das komplette Band sequentiell gelesen, sondern es kann zu der entsprechenden Stelle gespult werden.

keit gegenüber den optischen Medien nicht konkurrenzfähig. Seit wenigen Jahren werden deshalb auch die aktuellen Medien, CD und DVD, genutzt, allerdings weniger zur endgültigen Archivierung, als vielmehr zur vorübergehenden Datenauslagerung, um Speicherplatz auf der entsprechenden Festplatte freizugeben. Diese eingeschränkte Nutzung liegt zum einen daran, dass die Lebensdauer digitaler Medien noch nicht genau bestimmbar ist und zum anderen, dass optische Speicher, pro Gigabyte gerechnet für Laufwerk und Medium, im Gegensatz zu Magnetbändern auch heute noch zu teuer sind (Transtec 2003, S. 367 f). Die dauerhafte Archivierung der Daten erfolgt somit weiterhin auf Magnetbändern. Durch diese Art der Datenablage ist keine durchgängig einheitliche Verzeichnisstruktur gegeben, die es ermöglichen würde, den vorhandenen Datenbestand nach speziellen Abfragekriterien zu durchsuchen. Als Grundlage für eine Datenrecherche dienen daher nur die einzelnen Listings²⁰ der Magnetbänder oder entsprechend fortgeführte Projektordner. Damit wird das Auffinden und der Zugriff auf gewünschte Informationen erheblich erschwert.

2.3 Dateizugriff - Problemdarstellung

Der Auswerteprozess, so wie in Kapitel 2.1.3 dargestellt, erfolgt dateibasiert, so dass für (fast) jeden Auswerteschritt eine neue Datei angelegt wird. Alle zusätzlichen Informationen, die die einzelnen Daten näher beschreiben (s. 3.2.1 Metadaten), wie z.B. Besonderheiten bei der Auswertung, verwendete Algorithmen/ Filter, der aktuelle Bearbeitungsstand, müssen daher entweder in zusätzlichen Dateien abgelegt oder in analogen Medien fortgeführt werden. Teilweise wird auch von der Methode Gebrauch gemacht, erfolgte Bearbeitungsschritte direkt im Dateinamen zu vermerken. Aufgrund notwendiger Auswertevorgänge und aufgrund dessen, dass gleiche Dateien unter verschiedenen Gesichtspunkten Verwendung finden, ist es unumgänglich, dass Dateien entsprechend in andere Verzeichnisse kopiert, oder auf andere Rechnersysteme übertragen und umbenannt werden. Eine korrekte Laufenthaltung dieser losen Verknüpfung von Daten und Metadaten wird damit erheblich erschwert. Manche Informationen, die schon im Format selbst abgelegt werden, wie zum Beispiel die Angaben im SURF-Format zum verwendeten Gerätetyp und zu bestimmten Schiffskonfigurationen, können bei Konvertierungen in andere Dateiformate nicht übernommen werden und stehen für weitere Prozessierungen nicht zur Verfügung. Metadaten einer Datei sind daher in der Regel nur implizit über das Format, den Namen oder den Ort der Ablage zu erhalten (Krocker 2002). Für eine umfassende und ausführliche Darstellung der Metadaten bleibt letztlich derzeit nur die Möglichkeit eines konsequent geführten Projektordners.

Die Tatsache, dass die beschreibenden Informationen einer Datei in einem analogen Medium gehalten werden, erschwert die Suche nach Datensätzen, die einem gewünschten Aus-

²⁰Verzeichnisübersicht von den auf einem Magnetband enthaltenen Daten

2.3. DATEIZUGRIFF - PROBLEMDARSTELLUNG

wahlkriterium genügen sollen, erheblich. Globale Abfragen in der Art Wer? hat Wo? Wann? Was? (s. 3.2.1 Metadaten) gemessen, sind somit nicht über den kompletten Datenbestand durchführbar, sondern nur auf in Frage kommende Dateien im Einzelnen anzuwenden. Dies setzt allerdings eine lückenlose Kenntnis aller vorhandenen Daten voraus, welche aber nur langjährigen Mitarbeitern zu eigen ist.

Ein weiteres Problem der Datenverwaltung ist, dass eine auf Magnetband archivierte Datei nicht durch eine aktuellere, ähnlichen Inhaltes, ersetzt werden kann. Die Folge davon ist: Jede Datei, an deren Inhalt Veränderungen vorgenommen wurden, muss an einer anderen Stelle erneut gespeichert werden. Dies führt nicht nur zu Redundanzen in der Datenhaltung, sondern erschwert auch die Fortführung der Metadaten und den Datenaustausch hinsichtlich der Aktualität. Das Speichern der Daten auf den neueren Medien wie CD und DVD erleichtert den Umgang dabei nur dahingehend, dass ein wesentlich schnellerer Zugriff auf den gewünschten Datensatz erfolgen kann. Das Zurückschreiben einer einzelnen Datei ist auch hier noch nicht praktikabel.

Die dateibasierte Verwaltung der Daten und die unzureichend verknüpften Metainformationen haben zur Folge, dass Interessenten verschiedener Arbeitsgruppen des AWIs keine Möglichkeit haben, sich über den aktuell vorhandenen bathymetrischen Datenbestand zu informieren und ihre Anfragen diesbezüglich zu konkretisieren. Alle anfallenden Datenrecherchen müssen daher von den Mitarbeitern der Bathymetrie selbst vollständig vorgenommen werden.

3 Konzeptionelle Anforderungen an ein neues System zur bathymetrischen Datenverwaltung

Mit der Weiterentwicklung des Hydrosweeps¹ und dem vermehrten Einsatz von Flachwasserecholoten auf ROVs und AUVs² wird in Zukunft das Datenaufkommen pro Zeiteinheit weiter wachsen. Dies hat, neben dem steigenden Aufwand zur Verwaltung und Archivierung der Daten, auch zur Folge, dass Datenrecherchen und -analysen zunehmend umfangreicher und zeitintensiver werden. Die konzeptionellen Grenzen des derzeitigen Systems zur Verwaltung von bathymetrischen Daten sind somit erreicht. Daher ist es notwendig eine Struktur zu schaffen, die es erlaubt alle vorhandenen Datensätze der Bathymetrie inhaltlich, als auch physikalisch miteinander zu verknüpfen.

Dazu soll im ersten Teil von Kapitel 3 die Einsatzmöglichkeit eines geographischen Informationssystems (GIS) anhand seiner Funktionalitäten und Eigenschaften und der sich aus dem vorhergehenden Abschnitt ergebenden Anforderungen an ein neues System zur Verwaltung bathymetrischer Daten untersucht werden. Der zweite und dritte Teil des Kapitels beschäftigen sich mit der begrifflichen Strukturierung der Daten hinsichtlich ihres Inhaltes und mit der Gegenüberstellung von zwei verschiedenen Möglichkeiten der Datenorganisation. Diese beiden Abschnitte dienen als Grundlage für die später notwendigen Entscheidungen zur Konzeption eines geeigneten Datenmodells.

3.1 Zielsetzung - ein GIS?

„Ein Geoinformationssystem ist ein rechnergestütztes System, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst,

¹Das Sensorupdate HDBE (High Definition Bearing Estimation) erlaubt durch die detailliertere Auswertung des reflektierten Sonarsignals eine höhere Auflösung der Meeresbodentopographie (Umstellung von 59 „hard-beams“ auf 240 „soft-beams“, s. Hydrosweep, S. 7)

²Remotely Operated Vehicle und Autonomous Underwater Vehicle

KAPITEL 3. KONZEPTIONELLE ANFORDERUNGEN AN EIN NEUES SYSTEM ZUR BATHYMETRISCHEN DATENVERWALTUNG

gespeichert, verwaltet, aktualisiert, analysiert und modelliert sowie alphanumerisch und grafisch präsentiert werden. (Bill und Fritsch 1991, S. 5)

Der wesentliche Vorteil eines GIS besteht in der Möglichkeit geographische Informationen mit Sachdaten zu verknüpfen. Mit geeigneten Analyse-Werkzeugen ist der Nutzer dadurch in der Lage, auch über inhaltliche Abfragen, gewünschte raumbezogene Informationen zu selektieren. Hinzu kommen weitere Werkzeuge, welche in einer graphischen Oberfläche integriert sind, um den vorhandenen Datenbestand gezielt zu bearbeiten und zu untersuchen (s. 4.3.1 ArcGIS Desktop).

Die Datenverwaltung eines GIS wird zumeist mit in einem Datenbanksystem realisiert. Dies ermöglicht zum Einen, die unterschiedlichen Daten über entsprechende Indizes miteinander in Beziehung zu setzen und gewährleistet zum Anderen, den gleichzeitigen Zugriff mehrerer Nutzer auf denselben Datensatz.

Aufgrund der genannten Vorteile, soll zur Verwaltung der bathymetrischen Daten ein Geoinformationssystem zum Einsatz kommen. Allerdings handelt es sich im vorliegenden Fall um ein sogenanntes Web-GIS (oder Internet-GIS), welches erlaubt, über Internet/Intranet-Anwendungen auf die gewünschten Daten zuzugreifen. Im Unterschied zum Web-Mapping beschränkt es sich nicht auf die bloße Darstellung von Kartendaten, sondern bietet zusätzlich typische GIS-Funktionalitäten. Die Grundlage dafür soll ein Datenbanksystem sein, das es ermöglicht, den vorhandenen bathymetrischen Datenbestand hinsichtlich zeitlichen, inhaltlichen und geographischen Aspekten zu untersuchen. Dazu ist der Einsatz einer graphischen Oberfläche („Viewer“) denkbar, die zum einen alle vorhandenen Daten mit geographischem Bezug visualisiert und zum anderen über eine gewisse GIS-Grundfunktionalität verfügt, die jedem Nutzer, auch ohne Kenntnis der zugrundeliegenden Datenorganisation, die Generierung notweniger Abfragen intuitiv ermöglicht. Neben den Funktionen des Zoomens und des Selektierens soll der *Viewer* auch ein entsprechendes Tool zum Erstellen von SQL-Anweisungen besitzen. Das Hinzuladen von weiteren geographischen Objekten (Punkte, Linien, Flächen) würde ebenso zur Funktionalität des *Viewer* gehören, wie das Speichern des aktuellen Bildschirmausschnitts in einem Bildformat, um zum Beispiel Trackplots³ zu erzeugen. Alle durch Abfragen generierten Ergebnisdatensätze sollten dabei so angelegt werden, dass entsprechende Roh- und Ergebnisdaten (s. 3.2.2 Roh- und Ergebnisdaten) direkt abrufbar sind, um sie entweder, soweit es das Format zulässt, zu visualisieren oder um sie auf der nutzereigenen lokalen Festplatte zu speichern; gleiches gilt für Projekt- und Metadaten.

Grundsätzlich muss das Datenbanksystem so konzipiert sein, dass es neben den bathymetrischen Daten, die auf der Polarstern aufgezeichnet wurden, auch die Messungen der beiden anderen Forschungsschiffe des AWIs, Uthörn und Heincke, integrieren kann. Die

³Darstellung aller Schiffsrouten eines ausgewählten Gebietes in kartenähnlicher Form

Ergebnisse weiterer Fachbereiche (Gravimetrie, Magnetik oder Seismik), sofern es interdisziplinäre Interpretationen erfordern, sollten ebenso verwaltet werden können, wie die Daten anderer Organisationen (z.B. GEODAS⁴). Außerdem darf der Einarbeitung möglicher Weiterentwicklungen der Messgeräte in das System nichts entgegenstehen.

Dieses Datenbanksystem, das auch über Intranet bzw. Internet abfragbar sein soll, um Mitarbeitern anderer Institutionen die Möglichkeit der Datenrecherche zu geben, soll keinesfalls in Konkurrenz zu den bereits vorhandenen Datenbanken PANGEA und PODAS (s. 2.2 Datenverwaltung) treten. Vielmehr ist es erwünscht, um Redundanzen in der Datenhaltung und die daraus resultierenden Probleme hinsichtlich der Datenaktualität zu vermeiden, dass die Systeme sich gegenseitig sinnvoll ergänzen und die Verwaltung des vorhandenen Datenbestandes vereinfachen.

3.2 Datentypen

Im Folgenden werden die zu verwaltenden Daten der Bathymetrie hinsichtlich ihres Inhaltes strukturiert und entsprechend den vier Gruppen der Roh-, Ergebnis-, Projekt- und Metadaten zugeordnet, mit dem Ziel, die Diskussion zur Konzeption eines Datenmodells in Kapitel 5.1 zu vereinfachen. Weiterhin wird in diesem Abschnitt besonders auf die Bedeutung und die Notwendigkeit von Metadaten hingewiesen.

3.2.1 Metadaten

Was sind Metadaten? Mit Metadaten⁵ (= „Über“-daten) werden jene Daten bezeichnet, die einen durch bestimmte geodätische Messverfahren ermittelten Datensatz inhaltlich charakterisieren und beschreiben. Dazu gehören neben den Angaben zur Qualität, räumlichen Ausdehnung und verwaltenden Institution auch Informationen zum Inhalt und zur Art des zugrundeliegenden Datensatzes (s. Standards, S. 22). Metadaten sollen demnach alle Informationen enthalten, die zur Entscheidungsfindung hinsichtlich Brauchbarkeit und Verwendungsmöglichkeit eines Datensatzes notwendig sind.

Prinzipiell sind Metadaten formatunabhängig und können daher je nach Anwendung in einem Textfile, einer HTML- oder XML-Datei gespeichert oder als direkter Datenbankeintrag abgelegt werden. Aufgrund der geringen Größe gegenüber den Daten, die sie beschreiben,

⁴Geophysical Data System - eine von dem National Geophysical Data Center (NGDC) in den USA entwickelte Datenbank, die neben geophysikalischen und aeromagnetischen Daten auch Informationen zu bathymetrischen Fächersonar-Messungen enthält.

⁵Die Klärung des Begriffes erfolgt in diesem Abschnitt in Bezug auf geographische Daten.

KAPITEL 3. KONZEPTIONELLE ANFORDERUNGEN AN EIN NEUES SYSTEM ZUR BATHYMETRISCHEN DATENVERWALTUNG

erleichtern Metadaten wesentlich den Informationsaustausch mit anderen Firmen und Institutionen. Dieser Austausch findet Anwendung in eigens dafür geschaffenen Internetplattformen, sogenannten „Catalog Services“, die durch Auswahl von Kategorien und Eingabe von Schlüsselwörtern eine inhaltliche Suche nach Metadaten ermöglichen (s. Abb. 3.1). Bekannte Catalog Services sind das „Geography Network“⁶ und das „NSDI (National Spatial Data Infrastructure) Clearinghouse“⁷.



Abbildung 3.1: Catalog Service: Austausch-Plattform für Metadaten im Internet

Mit der Anlage und dem Austausch von Metadaten werden also grundsätzlich Informationen zu existierenden Daten für jeden, der danach sucht, zur Verfügung gestellt. Dies vereinfacht nicht nur erheblich das Auffinden, sondern auch die Interpretation und Anwendung der entsprechenden Daten (ESRI 2002c).

Wozu Metadaten? Die Arbeitsgruppe Bathymetrie des AWIs hat, wie alle Einrichtungen die sich mit geodätischen Problemen auseinandersetzen, unter anderem das Ziel, die

⁶vgl. <http://www.geographynetwork.com>; Datum des Zugriffs: 16.02.2004

⁷vgl. <http://www.fgdc.gov/clearinghouse/clearinghouse.html>; Datum des Zugriffs: 16.02.2004

Lage von Punkten im Raum zu bestimmen. Dabei kommen nicht nur verschiedene Messverfahren und -geräte zum Einsatz (s. 2.1.2 Tiefenmessung), sondern es sind besonders im Nachgang der Messungen Berechnungen und Analysen erforderlich. Ein nicht zu unterschätzender Teil der Auswertung ist dabei nach wie vor durch individuelle Handarbeit zu leisten, um die aufgezeichneten Daten entsprechend den Anwendungen zu bereinigen, zu bearbeiten und auch zu bewerten (s. 2.1.3 Datenfluss/ Formate).

Diesen Wert der aufgezeichneten Daten (Polarsternfahrt, Messgeräte, Personal) und den Wert der umfangreich getätigten Nacharbeit (Software, Arbeitskraft) zu erhalten, gelingt nur, wenn die daraus entstandenen einzelnen Produkte mit detaillierten zusätzlichen Informationen, also mit Metadaten, zur Art ihrer Entstehung versehen werden. Die Beschreibung der gesamten Auswertung ist derart anzufertigen, dass die Reproduktion der Produkte anhand der aufgezeichneten Daten durch einen Mitarbeiter der Arbeitsgruppe gegebenenfalls möglich ist. Dazu gehören neben den Angaben zur verwendeten Software und den angewandten Algorithmen und Filtern auch Informationen zu den Schwerpunkten und Überlegungen unter denen die Auswertung erfolgte, sowie eine geordnete Auflistung der vorgenommenen Bearbeitungsschritte. Von besonderer Wichtigkeit sind gerade die Sachverhalte, die während der Auswertung als selbstverständlich und offensichtlich gelten, wie zum Beispiel Dateistrukturen, notwendige Konvertierungen oder verwendete Passwörter, da sie nach Veränderung der zugrundeliegenden Software an Bedeutung verlieren und für nachfolgende Mitarbeiter nicht mehr direkt einsichtig sind. Die Metadaten sichern somit nicht nur die Qualität der Produkte, sondern auch deren langjährige Verwendung.

Sollte allerdings aus Nachlässigkeit oder Geringschätzung das Verfassen von Metadaten unterbleiben, verlieren bei einem personellen Wechsel alle nicht dokumentierten Daten ihren Wert, da eine Qualitätsangabe nicht unmittelbar von diesen ableitbar ist. Um die Daten dennoch verwenden zu können, wäre eine umfangreiche und zeitintensive Nacharbeit erforderlich.

Metadaten dienen innerhalb der eigenen Arbeitsgruppe in erster Linie der Qualitätssicherung und der Werterhaltung. Zudem verschaffen sie einen Überblick über den vorhandenen Datenbestand, der es erlaubt, Entscheidungen hinsichtlich notwendiger Folgemessungen, sei es zur Genauigkeitssteigerung oder zur Erweiterung des Messgebietes, fundierter zu treffen. Aber auch im Austausch mit anderen Organisationen und Institutionen sind Metadaten unerlässlich. Neben der Klärung rechtlicher Belange, bieten sie die Möglichkeit Daten anderer Einrichtungen mit den eigenen zu vergleichen, auf Brauchbarkeit hin zu untersuchen und gegebenenfalls zu übernehmen. Dadurch werden Mehrfachmessungen vermieden und Kosten gespart.

Aufgrund ihrer enormen Bedeutung für die Verwaltung und Verwendung raumbezogener Daten werden Metadaten in Zukunft daher nicht mehr lästiger Anhang, sondern fester Bestandteil eines jeden Messprojektes sein.

Der eigentliche Austausch von geographischen Daten erfolgt mit Hilfe von Standards. Die

KAPITEL 3. KONZEPTIONELLE ANFORDERUNGEN AN EIN NEUES SYSTEM ZUR BATHYMETRISCHEN DATENVERWALTUNG

zwei bedeutendsten Standards für Metadaten sollen im Folgenden kurz vorgestellt werden. Ein direkter Vergleich findet nicht statt.

Standards Metadaten sind die Grundlage für den Austausch von geographischen Informationen zwischen verschiedenen Institutionen. Damit sie aber zum einen in den sogenannten „Catalog Services“ veröffentlicht und zum anderen von Interessenten gesucht und gefunden werden können, ist es notwendig, deren Inhalt und Struktur zu standardisieren. Die zwei bedeutendsten Organisationen, die Standards für Metadaten entwickelt haben, sind das nationale Federal Geographic Data Committee (FGDC) der USA mit dem Content Standard for Digital Geospatial Metadata (CSDGM) und die International Organization for Standardization (ISO) mit der ISO-Norm 19115.

Das FGDC ist ein Zusammenschluß von Vertretern aus den Abteilungen und Ministerien der USA, die raumbezogene Informationen verarbeiten, mit dem Ziel, die gemeinsame Entwicklung, Verbreitung und Verwendung, sowie den Austausch von geographischen Daten zu fördern (FGDC 1998). Dies soll unter Einbeziehung von staatlichen Organisationen, Firmen und Universitäten mit der Schaffung einer Daten-Infrastruktur (NSDI) erreicht werden. Der derzeit aktuelle Standard für Metadaten des FGDC ist der CSDGM Version 2 (FGDC-STD-001-1998). Da dieser auch in der vorliegenden Arbeit zur Anwendung kommt, erfolgt hier eine nähere Erläuterung hinsichtlich des Aufbaus und Inhaltes.

Der Standard ist hierarchisch organisiert und setzt sich aus Containern oder Verbänden („Compound Elements“) und Daten („Data Elements“) zusammen, wobei erstere wieder durch weitere Verbände und Daten beschrieben werden können, während letztere jeweils elementar sind. Der Ausgangspunkt in der Hierarchie ist „Metadata“. Darin sind die folgenden Container enthalten, die wiederum durch die in Klammern angegebenen Verbände näher gekennzeichnet werden⁸:

- Identification (General, Contact, Citation, Time Period, Status, Spatial Domain, Keywords, Graphic, Security, Cross Reference)
- Data Quality (General, Attribute Accuracy, Positional Accuracy, Source Information, Process Step)
- Data Organization (General)
- Spatial Reference (General, Horizontal Coordinate System, Vertical Coordinate System)
- Entity Attribute (Detailed Description, Overview Description)

⁸Eine vollständige Übersicht des CSDGM befindet sich in FGDC (1998) oder im Metadaten-Editor des ArcCatalogs (s. 4.3.1 ArcGIS Desktop).

- Distribution (General, Distributor, Standard Order Process, Available Time Period)
- Metadata Reference (General, Extension)

Diese Hierarchie bietet nun die Möglichkeit alle Informationen, die notwendig sind, um einen Datensatz detailliert und umfassend zu beschreiben, in einer strukturierten und einheitlichen Anordnung abzulegen. Zudem ist, wie aus dem letzten Anstrich ersichtlich wird, dieser Standard so konzipiert, dass firmenspezifische Erweiterungen dennoch möglich sind. Dies nutzt zum Beispiel die Firma ESRI, indem sie mit einem eigenen Profil (ESRI Metadata Profile)⁹ vorhandene Datenstrukturen ausführlicher beschreibt.

Der ISO Metadaten Standard umfasst zwar inhaltlich die gleichen Informationen, ist aber strukturell etwas anders aufgebaut. Tendenziell gehen aber die Bestrebungen der beiden Organisationen dahin, die Standards anzugleichen, um den Austausch von Metadaten untereinander zu gewährleisten¹⁰.

Als Scriptsprache wird bei beiden Standards XML (Extended Markup Language) verwendet.

3.2.2 Roh- und Ergebnisdaten

Um im weiteren Fortgang der vorliegenden Arbeit, die in Abschnitt 2.1.3 beschriebenen verschiedenen Datenformate und Dateiformate nicht immer im Einzelnen zu benennen, ist es erforderlich, sie hinsichtlich ihres Inhaltes zu strukturieren.

Die Datenerfassung mit Sonaranlagen hat zum Ziel, mit Hilfe der ermittelten Meerestiefen, digitale Geländemodelle des Meeresbodens zu berechnen und bathymetrische Karten zu erzeugen. Alle Formate und Dateien die daher direkt zu diesem Zweck verwendet werden können, d.h. in denen eine direkte Zuordnung von Tiefen zu Positionen (x,y oder lat, long) besteht, sollen fortan mit Ergebnisdaten bezeichnet werden. Das sind neben den ASCII-Dateien, auch das HYD- und NAK-Format, in denen lediglich einige wenige zusätzliche Parameter, wie Zeit und Gyro¹¹, enthalten sind. Dagegen gehören alle Formate, die noch Angaben zur Messanlage und zur Messanordnung beinhalten, zur Gruppe der Rohdaten¹². Dazu zählen das SURF, sowie das DUX- und ROH-Format.

⁹vgl. <http://www.esri.com/metadata/esriprof80.html>; Datum des Zugriffs: 17.02.2004

¹⁰Informationen zur Harmonisierung von ISO 19115 und CSDGM Version 2 unter <http://www.fgdc.gov/metadata/whatsnew/fgdciso.html>; Datum des Zugriffs: 17.02.2004

¹¹Richtungswinkel bzgl. geographisch Nord

¹²Rohdaten sind in diesem Zusammenhang nicht gleichzusetzen mit unbearbeiteten Daten.

3.2.3 Projektdaten

Alle Folgeprodukte, die sich aus den Ergebnisdaten ableiten lassen und die im Rahmen von Diplom-, Promotions- oder Projektarbeiten entstanden sind, werden der Gruppe der Projektdaten zugeteilt. Dazu zählen vor allem, neben den Veröffentlichungen selbst, die in ArcInfo berechneten digitalen Geländemodelle und die daraus abgeleiteten bathymetrischen Karten. Die mit GMT (Generic Mapping Tools) erstellten Rasterbilder (NetCDF) gehören ebenso in diese Gruppe, wie auch zum Beispiel die mit dem Programm Fledermaus generierten Flugsequenzen.

3.3 Art der Datenablage

Die Aufgaben eines GIS sollten sich nicht nur darauf beschränken, die gestellten Abfragen zu bearbeiten, indem die dazu vorhandenen Informationen in einer Tabelle angezeigt werden, sondern es sollte außerdem dem Nutzer die Möglichkeit bieten, gewünschte Daten auf die eigene, lokale Festplatte in Dateiform zu speichern. Um dies zu realisieren gibt es, abhängig von der Art der Datenablage, prinzipiell zwei verschiedene Lösungswege, die im Folgenden, anhand ihrer Vor- und Nachteile, miteinander verglichen werden sollen:

1. sämtliche Daten sind in einzelnen Zeilen und Spalten von Tabellen abgelegt und werden von einem aufzurufenden Java-Programm¹³ ausgelesen und in eine Datei geschrieben
2. die Daten liegen bereits in Dateiform auf einem Fileserver vor und sind über einen Link abrufbar

Ein bedeutender Vorteil der ersten Variante besteht darin, dass alle Daten in einem einheitlichen Ablagesystem vorliegen. Somit sind Abfragen auf den Datenbestand jeglicher Art uneingeschränkt generierbar. Außerdem ist es dadurch möglich, beim Speichern von Datensätzen auf der lokalen Festplatte, nur wirklich die Datensätze mit Hilfe des Java-Programms auszulesen und in eine Datei zu schreiben, die den gewünschten Anforderungen entsprechen. So lassen sich zum Beispiel Daten, die entweder in einem Untersuchungsgebiet liegen oder ein markantes Merkmal aufweisen, gezielt extrahieren. Dies kann in diesem Umfang durch die zweite Variante keinesfalls geleistet werden, da jede einzelne Datei auf dem Fileserver in geschlossener Form vorliegt und daher keine inhaltlichen Abfragen und Selektionen zulässt und zudem nur begrenzte Verknüpfungsmöglichkeiten über einige wenige

¹³Java-Programm steht hier als Synonym für ein Programm, das die Möglichkeit bietet, aus einer Internetanwendung heraus, auf Datenbanken und Festplatten zuzugreifen.

Merkmale bietet. Dementsprechend werden beim zweiten Fall immer mehr Datensätze heruntergeladen als eigentlich erforderlich wären, was zur Folge hat, dass sie im Nachhinein noch auf Brauchbarkeit und Anwendbarkeit zu untersuchen sind.

Nachteilig wirkt sich dagegen bei der ersten Variante aus, dass zunächst eine Datenbankstruktur geschaffen werden muss, die es erlaubt, alle vorhandenen Formate zu verwalten. Die vielen verschiedenen, zum Teil auch dynamischen Dateistrukturen, erschweren das Aufstellen dieser Struktur. Außerdem müssten auf der einen Seite Algorithmen zum Einlesen von Dateien unterschiedlichen Formates programmiert werden, die die Datenintegrität und -konsistenz der Datenbankstruktur gewährleisten, und auf der anderen Seite Algorithmen zum Auslesen von Dateien, die deren Weiterverarbeitung in entsprechenden Anwendungsprogrammen ermöglichen. Der Aufwand zur Realisierung dieser Variante ist demnach gegenüber der Anlage und Verwendung eines Fileservers übermäßig hoch. Denn nach Fall 2 liegen alle Dateien in einem definierten Format auf dem Server vor und können entweder durch aktuellere leicht ersetzt oder nach dem Speichern auf der lokalen Festplatte mit einer entsprechenden Anwendung bearbeitet werden.

Keine der beiden beschriebenen Varianten bietet jeweils für sich genommen einen geeigneten Lösungsweg. Erst aus der Kombination der beiden Varianten lässt sich daher ein System erstellen, das die optimale Verwaltung des aktuellen bathymetrischen Datenbestandes gewährleistet und den in Abschnitt 3.1 gestellten Anforderungen entsprechen kann (s. 5.1 Konzeption des Datenmodells).

4 Unterstützende Software und Systeme

Die im Folgenden vorgestellten und zur Anwendung gekommenen Arbeitsplattformen und Softwareprodukte waren bis auf den Webserver bereits zu Beginn der vorliegenden Arbeit am Alfred-Wegener-Institut vorhanden. Sie werden im weiteren Verlauf entsprechend auf ihre Einsatz- und Anwendungsmöglichkeiten hin untersucht. Ein Vergleich mit ähnlichen Softwareprodukten findet dabei nicht statt.

4.1 Windows/ Internet

Die Ausführung der notwendigen Arbeitsschritte zur Umsetzung der in Abschnitt 3.1 beschriebenen Zielstellung erfolgte auf einer Windows 2000 Plattform (BATP8¹). Sie wurde gegenüber dem ebenfalls vorhandenen Solaris-System gewählt, weil zum einen nur sie die volle Funktionsfähigkeit des ArcIMS 4.01² (s. 4.3.3 ArcIMS) hinsichtlich der Internetanwendung gewährleistet und zum anderen, ein Windows-Rechner mit den entsprechenden Zugriffsrechten im AWI-Netzwerk einfacher zu administrieren (Installieren von Software, Verwalten von Diensten) ist. Zudem erleichtert die durchweg graphische Oberfläche des Windows-System eine schnellere Einarbeitung in neue Anwendungsprogramme (ArcGIS, ArcIMS).

Wie in Abschnitt 3.1 beschrieben, sollen auch Datenbankabfragen über das Intranet/ Internet möglich sein. Dazu bietet sich die Realisierung einer 3-Ebenen Client/ Server Architektur an (s. 4.3.3 ArcIMS). Als erforderliche „back-end“-Komponente wurde das bereits vorhandene Datenbanksystem Microsoft SQL Server genutzt.

Die Einrichtung der notwendigen „middleware“, die die Funktionalität des Dialogs zwischen

¹Name des lokalen Rechners im AWI-Netzwerk, auf dem alle notwendigen Installationen und Arbeitsschritte für die vorliegende Arbeit vorgenommen wurden.

²Die Installation eines ArcIMS ArcMap Servers und das Betrachten einer Webseite mit dem Java Custom Viewer ist nur auf einem Windows-System möglich.

Client und Datenbank bestimmt, war in Form des Internet Mapservers (ArcIMS) dagegen noch auf dem lokalen Rechner B ATP8 vorzunehmen. Folgende Systemvoraussetzungen waren dafür Bedingung:

- Vorinstallation des Service Pack 2, 3 oder 4 für Windows 2000
- 256 MB RAM für jeden Prozessor im Server
- 478 MB auf einer NTFS-Festplatte für die Installation, sowie mindestens 300 MB Auslagerungsspeicher
- Java Runtime Environment (JRE) mit Java Plug-in Version 1.3.1_02 oder Version 1.4.0 ist notwendig, empfohlen wird Java Development Kit (JDK) 1.4.0

Zusätzlich ist ein Webserver mit entsprechender Servlet Engine erforderlich, im vorliegenden Fall Apache 2.0.42 mit Tomcat 4.1.12³, der die Anfragen des Clients entgegennimmt, bearbeitet und an den ArcIMS weiterleitet.

An den Client selbst werden neben dem Vorhandensein eines Microsoft Internet Explorers Version 5.0 (oder höher) für die volle Kompatibilität keine weiteren Anforderungen gestellt. Eventuell notwendige Zusatzkomponenten können beim erstmaligen Öffnen einer mit ArcIMS erstellten Webseite noch nachträglich installiert werden. Die Verwendung des Netscape Communicators ist dagegen nur mit Einschränkungen möglich, da softwarebedingt nicht alle ArcIMS-Viewer unterstützt werden (s. 4.3.3 ArcIMS).

Auch eine Programmierumgebung war erforderlich, um das Konvertieren der Daten, die in die Datenbankstruktur eingelesen werden sollen, mit selbstgeschriebenen Programmen zu erleichtern. Dazu wurde Visual C++ 6.0 installiert.

4.2 Microsoft SQL Server 2000

Microsoft SQL Server 2000 ist ein relationales Datenbank-Managementsystem (RDBMS). Mit einer relationalen Datenbank ist die Möglichkeit gegeben Daten in Datenbanktabellen, die über gemeinsame Felder miteinander in Beziehung („Relation“) gebracht werden können, zu organisieren. Dazu stellt das Datenbank-Managementsystem (DBMS) eine Reihe von Werkzeugen zur Verfügung, die zum einen die Durchführung von Aufgaben im Hinblick auf Konfiguration, Verwaltung und Wartung des Servers erlauben und zum anderen das Einlesen, Speichern, Abfragen und Wiederfinden von Daten ermöglichen. Diese Werkzeuge

³Die notwendigen Systemanforderungen sind in den oben genannten bereits enthalten. Eine ausführliche Installationsanleitung ist zu finden unter <http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=24171> (Datum des Zugriffs: 08.12.2003)

bilden die Grundlage für die Datenbankschnittstelle ArcSDE (s. 4.3.2 ArcSDE), die den Datenaustausch zwischen der Anwendungssoftware ArcGIS sowie ArcIMS und dem Microsoft SQL Server 2000 realisiert. Da ArcSDE die vollständige Verwaltung (Anlage von Tabellen, Vergabe von Indizes) der raumbezogenen Daten übernimmt, ist es für den Anwender nicht notwendig, auf einzelne Funktionen und Eigenschaften der im Hintergrund arbeitenden Datenbank steuernd einzugreifen. Deshalb erfolgt in der vorliegenden Arbeit auch keine umfassende Beschreibung des relationalen Datenbanksystems, sondern es wird lediglich an den entsprechenden Stellen, soweit es für das Verständnis erforderlich ist, auf Besonderheiten des SQL Servers hingewiesen⁴.

Der SQL Server 2000 befindet sich physisch gesehen zusammen mit der ArcSDE auf dem Windows 2000 Server („db1.dmawi.de“) im Rechenzentrum des AWIs. Dort wird er entsprechend von den Datenbankadministratoren gewartet und konfiguriert, da er neben der Bathymetrie auch anderen Arbeitsgruppen als RDBMS zur Verfügung steht.

4.3 ArcGIS-Produktfamilie

ArcGIS ist der Name für ein Softwarepaket der Firma ESRI (Environmental System Research Institute) zur Verarbeitung (Verwaltung, Analyse, Modellierung und grafische Präsentation) raumbezogener Daten, das sich aus einzelnen aufeinander abgestimmten Komponenten zusammensetzt. Dabei wird prinzipiell zwischen Servern und Clients unterschieden (s. Abb. 4.1). Zu den Servern gehört ArcSDE, als Schnittstelle zu relationalen Datenbanksystemen, um Geodaten zentral zu verwalten und einer Vielzahl von Nutzern parallel zugänglich zu machen, sowie ArcIMS, mit der Funktion, raumbezogene Daten im Intranet oder Internet in Form digitaler Karten und interaktiver Anwendungen zur Verfügung zu stellen. Auf der Seite der Clients gibt es die ArcGIS Desktop Produkte ArcView, ArcEditor und ArcInfo mit unterschiedlichen Funktionalitäten⁵ zur Analyse, Modellierung und grafischen Darstellung von raumbezogenen Daten. Ebenso zählen die browserbasierten Anwendungen (HTML- und Java-Viewer) zur Visualisierung der mit ArcIMS erstellten Webseiten, sowie die Geodatenviewer ArcReader und ArcExplorer zur Gruppe der Clients. Alle Datentypen, die ArcGIS unterstützt, wie zum Beispiel Coverage, Shapefile, GRID, TIN und Imageformate können wahlweise in einzelnen Dateien oder in einer sogenannten *Geodatabase* (Geographic Database) in einem RDBMS gespeichert werden. Der Vorteil einer *Geodatabase* (s. 4.3.2 ArcSDE) besteht vor allem in der einheitlichen zentralen Verwaltung und den sich daraus ergebenden Möglichkeiten zur Nutzung und Aktualisierung der Daten. Weiterhin bietet ArcGIS ein Vielzahl von Erweiterungen (Spatial-, 3D-,

⁴Für eine ausführliche Beschreibung der einzelnen Funktionen und Eigenschaften des Servers s. Garcia et al. 2000.

⁵funktionale Gegenüberstellung der drei ArcGIS Desktop Produkte in ESRI 2003b, S. 14ff

KAPITEL 4. UNTERSTÜTZENDE SOFTWARE UND SYSTEME

Geostatistical-, Tracking Analyst, ArcPress, ArcScan) zur gezielten Bearbeitung von Geodaten sowie ein komplett zugängliches Objektmodell, welches eine individuelle Anpassung der einzelnen Komponenten erlaubt.

Im Folgenden soll nun noch auf diejenigen Anwendungen von ArcGIS näher eingegangen werden, die in der vorliegenden Arbeit auch zum Einsatz kommen. Eine ausführliche Beschreibung der anderen Komponenten und Erweiterungen ist in ESRI (2003b) zu finden.

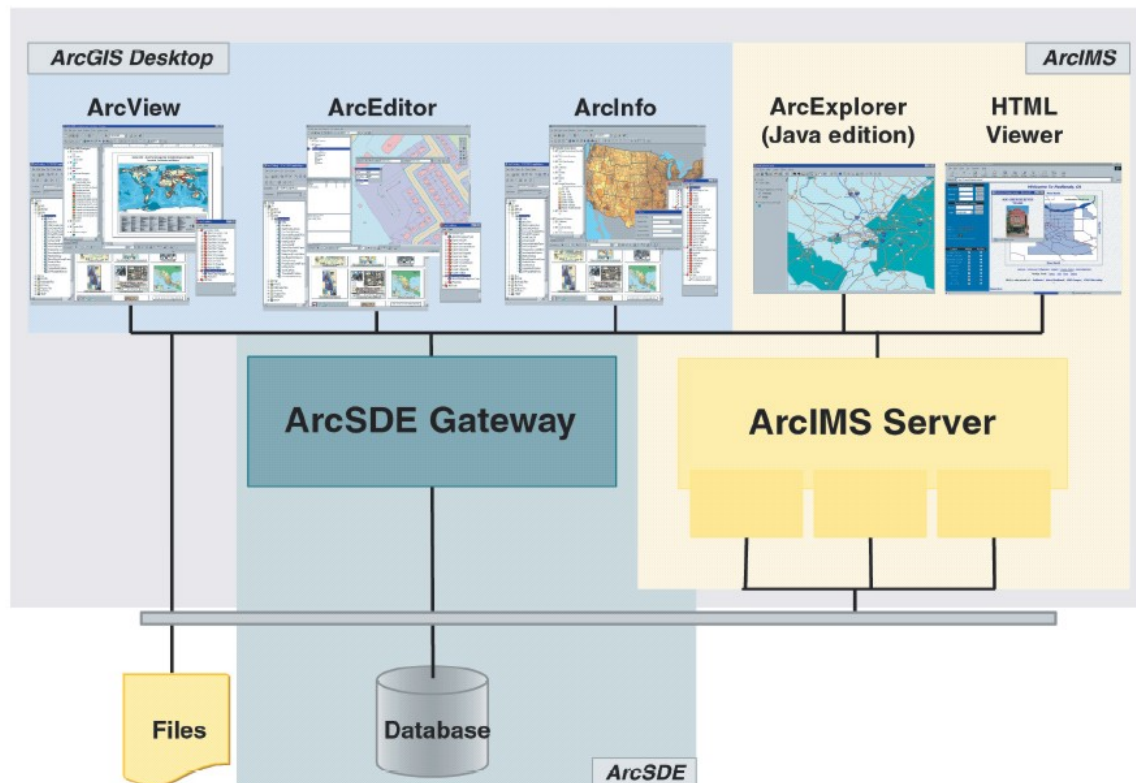


Abbildung 4.1: Die ArcGIS-Produktfamilie (ESRI (2002d), S. 2)

4.3.1 ArcGIS Desktop

ArcGIS Desktop ist der Überbegriff für die drei sich sinnvoll ergänzenden Anwendungen ArcCatalog, ArcMap und ArcToolbox, die zusammen alle Funktionen enthalten, die zur Erstellung und Fortführung eines GIS notwendig sind. Das umfasst sowohl das Konvertieren und Strukturieren als auch die Analyse und Darstellung geografischer Daten. Dabei spielt es für den Nutzer keine Rolle, ob die Daten in Dateien oder in einem Datenbanksystem organisiert sind, da der Zugriff und die Handhabung jeweils gleich ist.

ArcCatalog In seinem Aufbau, seiner Funktionsweise und seiner Handhabung ähnelt der ArcCatalog sehr dem Windows-Explorer, allerdings mit dem Unterschied, dass er speziell für die Strukturierung und Verwaltung von Geodaten geschaffen wurde. So erscheinen Datentypen, die sich aus mehreren Dateien zusammensetzen (zum Beispiel ESRI Shapefiles oder georeferenzierte TIFFs), im ArcCatalog dennoch nur als eine einzelne Datei. Dadurch lassen sich nicht nur Standardfunktionen wie Umbenennen, Verschieben und Kopieren auf diese Formate leichter anwenden, sondern es dient auch einer verbesserten Übersichtlichkeit, da wesentlich weniger Dateien überhaupt erst angezeigt werden. Darüber hinaus kann der Inhalt jedes Datensatzes im „Preview“ in tabellarischer oder grafischer Form betrachtet werden.

Eine besonders wichtige Funktion des ArcCatalogs ist der Aufbau von Verbindungen zu den im Hintergrund arbeitenden Datenbanken. Erst dadurch wird der eigentliche Zugriff gewährleistet und das Anlegen einer *Geodatabase* ermöglicht. Dazu stehen noch eine Reihe weiterer Werkzeuge zur Verfügung, die die Eigenschaften, Attribute und Wertebereiche der einzelnen Datenebenen und deren Beziehungen untereinander definieren. Zudem gibt es sowohl Funktionen für den Datenimport und -export als auch für die Organisation und Verwaltung der *Geodatabase* an sich.

In ArcCatalog ist außerdem die Metadatenverwaltung integriert, die es erlaubt, für jeden Datensatz, sei es nun eine einzelne Datei oder die Datenebene einer *Geodatabase*, Metadaten zu generieren. Dies erfolgt entweder nach einem der vorhandenen Schemata (ISO, FGDC, ESRI) oder nach eigenen Vorgaben. Die Metadaten in ArcCatalog können jederzeit angezeigt, bearbeitet oder nach geographischen, zeitlichen und inhaltlichen Aspekten durchsucht werden.

ArcMap ArcMap ist die zentrale Anwendung innerhalb von ArcGIS Desktop. Es enthält alle Werkzeuge, die zur Erstellung, Bearbeitung und Ausgabe von Karten jeglicher Art notwendig sind. Zudem verfügt es über verschiedene Funktionsgruppen und optionale Erweiterungen wie „Spatial Analyst“ oder „3D Analyst“, die eine umfangreiche Analyse des jeweils aktuellen Datenbestandes erlauben. Die eingelesenen bzw. importierten Daten werden in einzelnen Ebenen verwaltet und sind somit den jeweils entsprechenden Erfordernissen anpassbar. Alle vorgenommenen Änderungen sowohl am Inhalt als auch am Layout der Daten, können in einem gemeinsamen ArcMap-Projektfile (*.mxd) gespeichert werden. Eine wichtige Funktion von ArcMap hinsichtlich der Datenbankanwendung ist die Verknüpfung zweier oder mehrerer Datenbanktabellen über gemeinsame Felder. Aufgrund dieser inhaltlichen Vereinigung ist es möglich, Informationen wesentlich konkreter und gezielter über mehrere Tabellen abzufragen. Die dabei selektierten Ergebnisdaten können in eine neue Datei exportiert werden und stehen somit weiteren Anwendungen direkt zur Verfügung.

Zudem enthält ArcMap Werkzeuge wie das „Versioning“ oder „Disconnected Editing“, die das gleichzeitige Arbeiten mehrerer Nutzer an demselben Datensatz gewährleisten, und die die Möglichkeit bieten, gewünschte geographische Ausschnitte aus der *Geodatabase* in eine lokale Datenstruktur zu exportieren und nachdem diese mit entsprechenden Daten ergänzt wurde schließlich wieder zu importieren. Dies findet zum Beispiel Anwendung in Fällen, in denen keine Netzwerkverbindung zur *Geodatabase* besteht (z.B. auf Expeditionen), aber dennoch auf deren Daten zugegriffen werden soll.

ArcToolbox In ArcToolbox sind alle Werkzeuge (vorwiegend Konverter) für den Datenaustausch zwischen den in ArcGIS verwendeten Datentypen enthalten. Dazu stehen für viele Werkzeuge zwei verschiedene Eingabemöglichkeiten zur Verfügung: zum einen die Eingabemaske, in der alle notwendigen Parameter einzutragen sind, und zum anderen der Assistent, der alle erforderlichen Eingaben erläutert und abfragt. Außerdem ermöglicht ArcToolbox die Generierung von Batch-Prozessen, was bedeutet, dass mehrere verschiedene Programmabläufe, nachdem sie einmal definiert sind, nacheinander ausgeführt werden können.

4.3.2 ArcSDE

Anwendungsbereich/ Funktion ArcSDE (Spatial Database Engine) ist die Schnittstelle zu den relationalen Datenbanken. Durch sie ist es möglich, raumbezogene Daten in einem Datenbanksystem zu speichern und zu verwalten. ArcSDE unterstützt die Arbeit mit einer Vielzahl verschiedener Datenbanksysteme wie Oracle, Informix, IBM DB2 oder Microsoft SQL Server und gewährt dennoch immer den gleichen Zugang zu den entsprechenden Geodaten. Dadurch ist es möglich mit den ArcGIS Desktop Produkten, aber auch anderen Clients, unabhängig von der zugrundeliegenden Datenbank, auf die geographischen Informationen zuzugreifen.

Durch ArcSDE wird nicht nur die zentrale Datenhaltung in einem DBMS ermöglicht, sondern auch die Leistungsfähigkeit bei der Verwendung von Massendaten gesteigert, sowie der Aufbau von Geodatenmodellen, wie der *Geodatabase*, unterstützt. In ihrer Funktion als Application Server gestattet ArcSDE die Anfragen mehrerer Nutzer und Anwendungen parallel zu bearbeiten und entlastet somit zusätzlich das DBMS (s. Abb. 4.2).

Geodatabase Die *Geodatabase* (Geographic Database) ist eine relationale Datenbank mit räumlichen Daten als Inhalt. Die Besonderheit besteht dabei in dem zugrundeliegenden objekt-relationalen Vektor-Datenmodell. In diesem Modell werden Entitäten⁶ als Objekte

⁶eindeutig identifizierbare Objekte der realen Welt - die Zeile einer Datenbanktabelle

mit Eigenschaften, Verhalten und Beziehungen dargestellt. Dadurch wird es möglich, diese einzelnen Objekte miteinander in Beziehung zu setzen und zusätzliche Strukturen wie Topologien, Wertebereiche oder Gültigkeitsregeln zu definieren. Diese Strukturen werden in der *Geodatabase* als „Vektoren“ abgebildet und lassen sich daher je nach Bedarf erstellen und verändern. Neben den „Vektoren“ können auch verschiedene Raster-Datensätze, TINs (Triangulated Irregular Network) und nutzer-definierte Objekte gespeichert werden (s. Anhang A.1).

Erst die *Geodatabase*, als Kombination aus relationaler Standarddatenbank und ArcSDE Businesslogik (ESRI 2002d, S. 12 ff), ermöglicht die Realisierung eines geographischen Informationssystems.

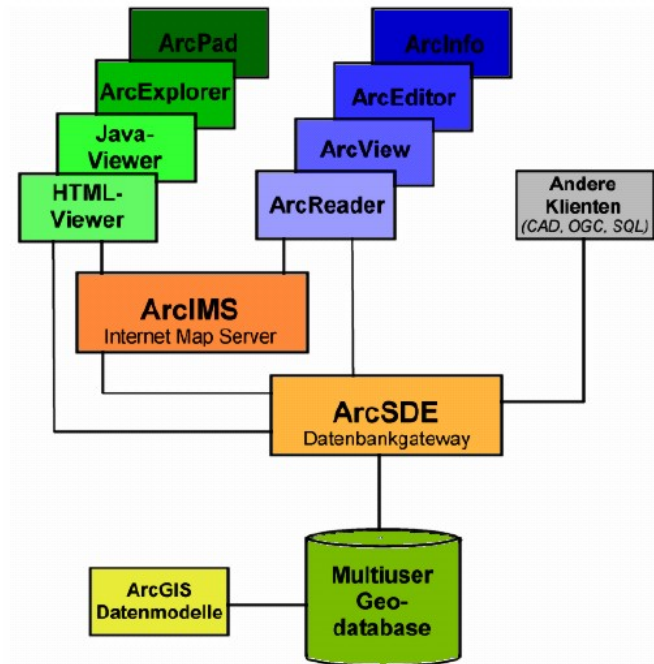


Abbildung 4.2: Die ArcSDE als Datenbankgateway(ESRI (2003b), S. 25)

Datenverwaltung Alle Daten werden in Standard-Datenbanktabellen, unter Verwendung der entsprechend verfügbaren Datentypen des zugrundeliegenden Systems, gespeichert. ArcSDE ersetzt keine bereits bestehenden DBMS Funktionalitäten, sondern nutzt und ergänzt stattdessen die jeweils vorhandene Leistungsfähigkeit, indem sie eine zusätzliche räumliche Komponente der Datenbank hinzufügt (ESRI 2002d, S. 26 ff).

Raumbezogene Daten werden von ArcSDE in sogenannten *Feature Classes* („Merkmalsklassen“) organisiert (s. Anhang A.1). Eine *Feature Class* ist eine Sammlung von einem oder mehreren *Features* eines geometrischen Types (Punkt, Linie, Fläche). Das *Feature* selbst ist das geometrische Abbild eines räumlichen Objektes (z.B. Straße) und wird als eine Folge von einem oder mehreren x-, y-Koordinatenpaaren und deren Eigenschaften zueinander definiert (ESRI 2002d, S. 26 ff). Die Geometrie eines jeden *Feature* wird in dem abstrakten geometrischen Datentyp „Shape“ abgelegt. Dadurch wird der direkte Zugriff auf raumbezogene Informationen ermöglicht. Außerdem legt ArcSDE für jedes *Feature* innerhalb einer *Feature Class* einen räumlichen Index an, um die Ausführung von Abfrage- und Analysefunktionen zu beschleunigen (s. Abb. 4.3).

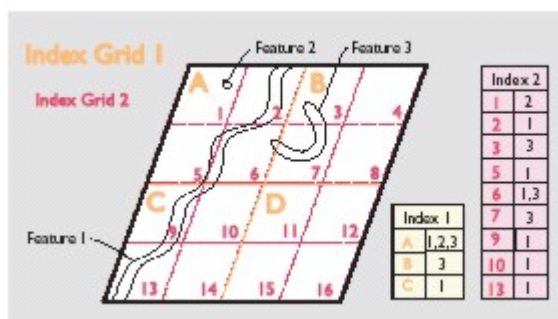


Abbildung 4.3: Anlage eines räumlichen Index zum schnellen Wiederfinden von raumbezogenen Daten(ESRI (2002d), S. 32)

Verbindungstypen Prinzipiell bietet ArcSDE zwei verschiedene Konfigurationen, um eine Verbindung zwischen der GIS-Anwendung und dem Datenbanksystem herzustellen. Zum einen gibt es die 3-Ebenen Architektur, unter Verwendung des ArcSDE Application Servers und zum anderen die 2-Ebenen Architektur, mit der sogenannten „direkten Verbindung“. Der Vorteil der zweiten Variante liegt in der besonders einfachen Administration und Installation, da der ArcSDE Server nicht installiert werden muss. Allerdings wird dadurch dessen Leistung auf die Desktop-Anwendungen verlagert, was zusätzliche Rechenkapazität des Desktop-Computers erfordert. Nachteilig bei diesem Verbindungstyp ist das erhöhte Datentransfervolumen über das Netzwerk.

Mit der 3-Ebenen Architektur übernimmt der ArcSDE Application Server die Bearbeitung der Anfragen der Clients und die entsprechende Formulierung der Antworten. Durch eine Vielzahl von Konfigurationsmöglichkeiten, um ArcSDE Server und DBMS aufeinander abzustimmen (ESRI 2002a), wird im Allgemeinen eine schnellere Durchführung dieser Prozesse als mit der „direkten Verbindung“ erreicht. Zusätzliche Funktionen des ArcSDE

Servers wie „data buffering“ und „cooperative processing“ ermöglichen zudem eine optimierte Netzauslastung. Ein Nachteil dieser Architektur ist der höhere Administration- und Installationsaufwand (ESRI 2002d, S. 36 f und ESRI 2003b, S. 25 f).

Im Rahmen der vorliegenden Arbeit erfolgte der Zugriff auf das DBMS über den ArcSDE Application Server. Dieser befindet sich physisch gesehen ebenfalls auf dem Windows 2000 Server („db1.dmawi.de“) und wird entsprechend vom Rechenzentrum administriert (s. 4.2 Microsoft SQL Server 2000).

4.3.3 ArcIMS

Anwendungsbereich/ Funktion Um anderen Arbeitsgruppen des AWIs und auch Mitarbeitern anderer Organisationen die Möglichkeit zu geben, sich über den vorhandenen Datenbestand der Bathymetrie zu informieren, ist es notwendig, diesen via Internet/ Intranet-Anwendungen zugänglich zu machen. Grundlage dafür ist die klassische 3-Ebenen Client/-Server Architektur (s. Abb. 4.4), bestehend aus Client (Presentation Tier – s. ArcIMS Viewer, S. 38), Middleware (Business Logic Tier – ArcIMS) und Datenbank (Data Tier – SQL Server mit ArcSDE). Die Kommunikation zwischen den einzelnen Ebenen erfolgt dabei über Anfrage/ Antwort-Paare (Request/ Response).

Die Middleware enthält alle Komponenten, die zum Starten eines Services und zum Verarbeiten der Anfragen und Antworten erforderlich sind. Dazu zählen neben den ArcIMS Produkten (Spatial Server, Application Server, Application Server Connector und Manager) auch der Webserver mit entsprechender Servlet Engine⁷. Sendet der Client nun eine Anfrage, wird diese vom Webserver empfangen und über die zugehörige Servlet Engine an den ArcIMS Servlet Connector weitergegeben. Der Servlet Connector übersetzt die Anfrage in eine ArcXML-Datei (Extensible Markup Language) und schickt diese zum ArcIMS Application Server, der sie an den Spatial Server weiterleitet. Dieser berechnet nach den Vorgaben der ArcXML-Datei die entsprechende Antwort und sendet sie über den umgekehrten Weg wieder zurück zum Client (ESRI 2003a, S. 20 und Kurtze 2003, S. 28 f).

ArcIMS Spatial Server Der ArcIMS Spatial Server ist das eigentliche „Rückgrat“ des ArcIMS, da er die angeforderten Karten und Daten in geeigneten Formaten zusammenstellt und sie lokal in einem Arbeitsverzeichnis speichert. Dem Client sendet er eine Antwort in Form einer ArcXML-Datei, die die Pfadangaben der entsprechend erstellten Ergebnisdaten enthält (s. Abb. 4.5). Der Spatial Server beinhaltet dazu verschiedene Servertypen, die

⁷Die Servlet Engine unterstützt die Verbindung zwischen dem Webserver und der Java Virtual Maschine (ESRI 2003a, S. 3). Diese wird notwendig, da viele Java 2 Komponenten in ArcIMS enthalten sind. Wie der Webserver ist die Servlet Engine nicht in ArcIMS enthalten und muss daher ebenfalls zusätzlich installiert werden (s. 4.1 Windows/ Internet).

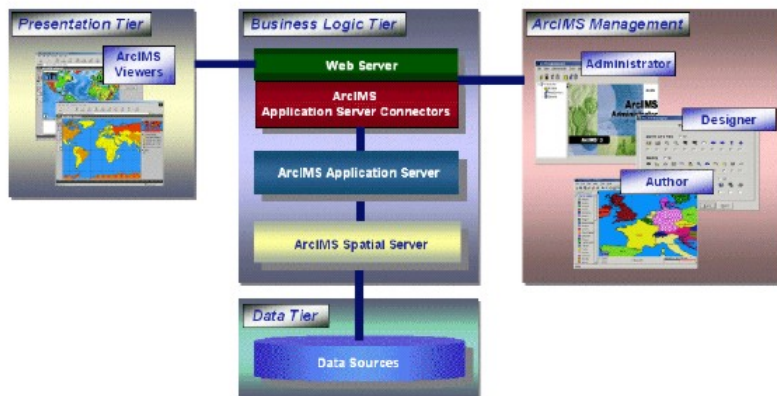


Abbildung 4.4: Architektur des ArcIMS (ESRI (2003a), S. 1)

sogenannten *Virtual Server*, die jeweils einen speziellen Anfragetyp bearbeiten können⁸ (s. Abb. 4.6).

- Der *Image Server* generiert für jede Anfrage des Clients eine Karte im Rasterformat (JPEG, GIF, PNG) und sendet diese zum Client zurück. Als Kartengrundlage dienen *Shapefiles* sowie alle von ArcSDE unterstützten Datentypen.
- Der *ArcMap Server* generiert ebenfalls Karten im Rasterformat. Er verwendet dagegen als Grundlage ArcMap-Projektfiles (*.mxd), wodurch es möglich ist, zusätzliche kartographische Gestaltungsmöglichkeiten in die entsprechenden Karten zu übernehmen.
- Vektordaten von ESRI Shapefiles und ArcSDE-Layern werden vom *Feature Server* in einem komprimierten Format zusammengestellt („Feature Streaming“) und zu einem Java-Applet in den Webbrowser des Clients gesendet. Durch dieses Format werden zusätzliche Funktionalitäten am Client ermöglicht. Dazu zählen u.a individuelle Kartengestaltung oder die Durchführung von Attributabfragen ohne erneuten Serverkontakt (s. *Image Server*).
- Der *Metadata Server* verwaltet die Dokumente, welche Metainformationen zu den einzelnen Daten und Karten enthalten. Er bearbeitet demnach alle Anfragen, die inhaltliche, geographische und zeitliche Aspekte der Metadaten betreffen.

⁸Um die Rechenlast optimal zu verteilen, besteht die Möglichkeit, mehrere Spatial Server, die sich auch auf unterschiedlichen Rechnern befinden können, parallel zu betreiben. Darüber hinaus kann auch jedem einzelnen *Virtual Server* entsprechend erforderliche Rechenleistung zugewiesen werden.

- Die Funktion des *Query Servers* besteht darin, Attributdaten abzufragen, die direkt mit den Shapefiles oder ArcSDE-Layern verknüpft sind.
- Der *Geocode Server* wird verwendet, um Punkte entsprechend ihrer Adressinformationen auf einer Karte zu lokalisieren („Geocoding Services“ in ESRI 2002b, S. 269 ff).
- Der *Extract Server* bietet die Möglichkeit, Daten aus Shapefiles und ArcSDE-Layern zu extrahieren und sie, im Unterschied zum *Feature Server*, in einem komprimierten Shapefile dem Client zum Download zur Verfügung zu stellen.

Diese *Virtual Server* werden in die beiden Kategorien „public“ (Image, ArcMap, Feature und Metadata Server) und „private“ (Query, Geocode und Extract Server) unterteilt. Während Erstere für den Nutzer über den ArcIMS Manager frei verfügbar sind, werden Letztere nur je nach Anwendung vom ArcIMS selbst aufgerufen.

A sample response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-87.5" miny="30.0" maxx="-59.5" maxy="50.0" />
      <OUTPUT file="C:\ArcIMS\output\usa_image_MYCOMPUTER2953026.jpg"
url="http://mycomputer.domain.com/output/usa_image_MYCOMPUTER2953026.jpg" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Abbildung 4.5: Der Antwort-Typ des Spatial Servers richtet sich nach der Art der gestellten Anfrage (in dieser Abbildung in Form eines JPG-Bildes). Prinzipiell gibt es neun verschiedene Antworttypen des Spatial Servers.

ArcIMS Application Server Der Application Server läuft als Hintergrundprozess und regelt die Verteilung der eingehenden Anfragen. Zudem registriert er, welcher Service auf welchem Spatial Server läuft, um die eingegangenen Anfragen entsprechend richtig weiterzuleiten. Der Application Server ist dabei so angelegt, dass er nur ArcXML-Dateien interpretieren kann. Dies bedingt zwar, dass zunächst alle Anfragen des Clients durch die Application Server Connectors in ArcXML übersetzt werden müssen, bietet aber gleichzeitig auch Nicht-ArcIMS Clients die Möglichkeit den MapService von ArcIMS zu nutzen.

ArcIMS Manager Der ArcIMS Manger⁹ stellt mit dem Author, Designer und Administrator drei Werkzeuge zur Verfügung, die es erlauben, eine internetfähige Webseite mit

⁹unter Windows NT/2000 auch als *Remote Service* verfügbar

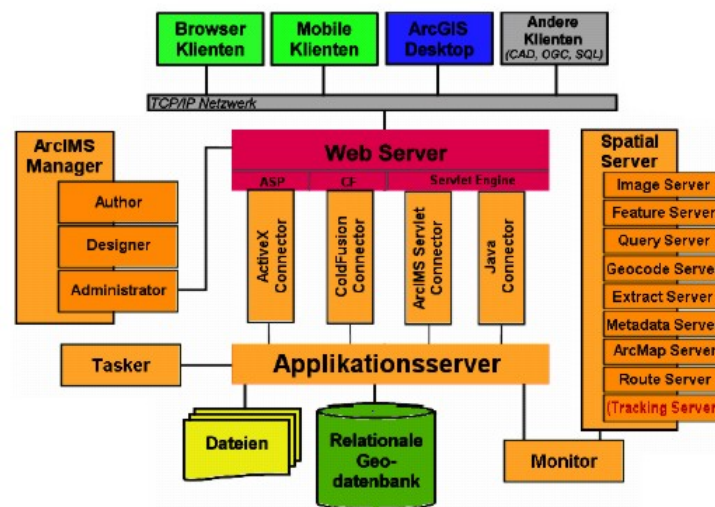


Abbildung 4.6: Die Systemkomponenten des ArcIMS

gewünschtem Karteninhalt und -layout zu erstellen und den entsprechend zugehörigen Service zu verwalten. Die Kommunikation zu ArcIMS erfolgt auch hier wieder über ArcXML. Zunächst wird mit dem Author oder ArcMap eine Kartenansicht erstellt und diese dann in einem sogenannten *Map Configuration File* (*.axl) oder in einem ArcMap-Projektfile (*.mxd) gespeichert. Auf der Grundlage dieses Files wird mit dem Administrator ein Service auf einem der gewünschten „public“-Server gestartet und administriert. Der Designer stellt verschiedene Optionen, in Abhängigkeit des gewählten Viewers, zur Verfügung, um den Service in eine internetfähige Webseite zu integrieren. Das Ergebnis ist ein Verzeichnis mit allen HTML- und JavaScript-Dateien, die notwendig sind, um diese Webseite in einem Browser aufzurufen, die erstellte Kartenansicht zu visualisieren und entsprechende Abfragen durchzuführen.

ArcIMS Viewer Der ArcIMS stellt innerhalb der Präsentationsebene (Presentation Tier, s. Abb. 4.4) mit dem HTML-Viewer und den beiden Java-Viewern (Standard und Custom) drei Clients zur Verfügung, die dem Nutzer die Möglichkeit bieten über Internet/Intranet-Anwendungen auf geographische Informationen zuzugreifen, sie zu visualisieren und gegebenenfalls zu bearbeiten. Sie enthalten dazu jeweils ein Kartenfenster und eine Werkzeugleiste zur interaktiven Bedienung. Die ArcIMS Viewer werden mit dem ArcIMS Designer bei der Generierung der Webseite erzeugt.

Der HTML-Viewer ist ein sogenannter „thin“-Client (Dickmann 2001, S. 59 f), da er keine clientseitigen Operationen mit dem Datenmaterial anbietet. Für jede Datenabfrage oder -selektion seitens des Nutzers wird daher eine neue Anfrage generiert und an den ent-

sprechenden Service gesendet. Die Antwort ist jeweils eine Karte im Rasterformat. Folglich kann der HTML-Viewer nur auf Karten und Sachdaten eines Image- oder ArcMap-Services zugreifen. Der Vorteil dieses Clients liegt darin, dass er vollständig aus HTML und JavaScript besteht. Damit besitzt er nicht nur eine hohe Anpassungsfähigkeit, sondern kann auch plattform-unabhängig eingesetzt werden (ab Microsoft Internet Explorer 5.0 oder Netscape 4.0).

Die Java-Viewer (Standard und Custom) verwenden zur Darstellung der Kartendaten ein in HTML eingebettetes Java2-Applet. Die Daten werden dabei vom ArcIMS statt im Rasterformat als Shapefile oder mit Hilfe des „Feature Streaming“ an den Client übertragen. Dies erlaubt dem Nutzer nicht nur Abfragen und Selektionen für den ausgewählten Bereich ohne erneuten Serverkontakt durchzuführen, sondern auch lokale Daten hinzuzuladen. Außerdem ist es möglich, mehrere Services (Image, ArcMap oder Feature) gleichzeitig in einem Java-Viewer darzustellen. Aufgrund seiner Funktionalität, gehört der Java-Viewer damit in die Gruppe der „thick“-Clients (Dickmann 2001, S. 59 f).

Allerdings wird durch die Verwendung des Java2-Applets die Anpassungsfähigkeit und die Browserkompatibilität der Java-Viewer eingeschränkt. Der Custom-Viewer erlaubt zwar im Gegensatz zum Standard-Viewer noch geringe nutzer-spezifische Veränderungen, kann jedoch nur noch vom Internet Explorer (Version 4.x oder höher) interpretiert werden.

5 Praktische Umsetzung

Zur Verwaltung der bathymetrischen Daten kommt, wie in der Zielsetzung (Abschnitt 3.1) beschrieben, ein web-basiertes geographisches Informationssystem zum Einsatz. Dieses Web-GIS soll den direkten Zugriff auf Ergebnis-, Roh-, Meta- und Projektdaten (s. 3.2 Datentypen) gleichermaßen gewährleisten, um zum einen die Aktualisierung und Fortführung der Daten für die Mitarbeiter der Bathymetrie zu vereinfachen (s. 2.2.3 Bathymetrische Datenverwaltung) und zum anderen, den Nutzern außerhalb der Arbeitsgruppe die Möglichkeit zu bieten, den vorhandenen Datenbestand nach inhaltlichen und geographischen Aspekten zu untersuchen.

Das Web-GIS basiert wie PANGAEA ebenfalls auf einer 3-Ebenen Client/Server Architektur, bestehend aus Datenbank, Middleware und Client. Allerdings soll im Unterschied zu PANGAEA (s. 2.2.2 PANGAEA) der Nutzer hier in der Lage sein, auf der Grundlage von graphisch Dargestelltem, Datenrecherchen durchzuführen, um einen besseren Überblick über die verfügbaren Daten zu erhalten und somit gezielter gewünschte Informationen abzufragen. Für die Umsetzung dieser 3-Ebenen Architektur sollen im Folgenden die zwei Komponenten 'Datenbank' und 'Client' näher betrachtet werden. Auf die 'Middleware' wird nicht erneut eingegangen, da diese bereits in Abschnitt 4.3.3 ausführlich erläutert wurde.

Während sich die ersten beiden Abschnitte des Kapitels mit der Konzeption und der Anlage des Datenverwaltungssystems beschäftigen, befassen sich die beiden letzten Abschnitte mit der Erstellung und Anpassung der graphischen Oberfläche, über die der Nutzer auf den bathymetrischen Datenbestand zugreifen kann. Dabei stehen jeweils die in der Zielsetzung (3.1) formulierten Anforderungen und deren Umsetzung im Vordergrund. Im mittleren Teil wird anhand von Beispieldatensätzen gezeigt, welche Schritte notwendig sind, um diese in das Web-GIS einzulesen und entsprechend untereinander und mit Metadaten zu verknüpfen. Dazu werden Daten aus dem Bereich rund um die Antarktis verwendet, d.h. südlich von 50 Grad südlicher Breite.

5.1 Konzeption des Datenmodells

Um einen optimalen Zugang zu den bathymetrischen Daten zu gewährleisten, sei es in Form von Abfragen oder in Form des direkten Zugriffes, werden die beiden in Abschnitt 3.3 vorgestellten Möglichkeiten der Datenablage miteinander kombiniert. Als Ergebnis dieser Diskussion werden die Daten, auf die über Abfragen und Selektionen zugegriffen werden soll, in der Datenbank gespeichert, während alle Dateien, deren Format einen geschlossenen Zugriff notwendig macht, auf dem Fileserver abgelegt werden. Die Verknüpfung zwischen Datenbank und Fileserver erfolgt dabei über den ArcIMS Viewer, der dazu noch entsprechend angepasst werden muss (s. 5.6 Projektorientierte Anpassung).

Allerdings ist bei der Organisation der Daten in Datenbanktabellen hinsichtlich der Internet/Intranet Anwendung zu beachten, dass in dem ArcIMS Viewer nur die Daten abgefragt werden können, die unmittelbar mit einem geographischen Objekt (Punkt, Linie, Fläche) verknüpft sind. Außerdem können Abfragen nicht über mehrere Tabellen gleichzeitig durchgeführt werden, wie das zum Beispiel in ArcMap möglich ist (s. ArcMap, S. 31). Daraus ergibt sich, dass nur die Ergebnisdaten von den vorhandenen Datentypen (s. 3.2 Datentypen) in der Datenbank gespeichert werden können, da sie über den direkten räumlichen Bezug verfügen und da ihr Format entsprechend einfach zu handhaben ist. Auf der Grundlage dieser Daten können somit bereits geographische Selektionen vorgenommen werden.

Um auch Abfragen inhaltlicher Art zu ermöglichen, ist es notwendig, die Ergebnisdaten mit ausgewählten zusätzlichen Informationen (wie z.B. Messplattform, Fahrtnamen, Schiffsnamen) zu verknüpfen. Damit diese Informationen aber nicht an jeden einzelnen Datensatz als Attribute angehängt werden müssen, wird eine Metadatentabelle ('Cruise') angelegt, die alle erforderlichen Angaben zum jeweiligen Fahrtabschnitt enthält und die über einen Index, wie in Abbildung 5.1 dargestellt, mit den entsprechenden Ergebnisdaten verknüpft ist. Der räumliche Bezug der Metadatentabelle wird mit Hilfe von Linien, die den Fahrtverlauf der jeweiligen Expedition widerspiegeln, hergestellt. Damit wird erreicht, dass die 'Cruise'-Tabelle ebenfalls im ArcIMS Viewer visualisierbar ist und dass deren Attribute für den Nutzer abfragbar sind. Die Metadatentabelle bildet somit die Grundlage für die Analyse des aktuellen bathymetrischen Datenbestandes hinsichtlich der Beantwortung der Fragen Wer? hat Wo? Wann? Was? gemessen (s. 2.3 Dateizugriff).

Da nicht immer alle Tiefenmesssysteme (DWS, Parasound und Hydrosweep, s. 2.1.2 Tiefenmessung), aufgrund von Systemausfällen und eisbedingten Störungen, über den gesamten Zeitraum eines Fahrtabschnittes im Einsatz waren, sollen die Stellen, an denen Tiefenmessungen erfolgten, durch Punkte im ArcIMS Viewer visualisiert werden. Dazu wird zunächst in Tabelle 5.1 für jedes Messsystem eine Abschätzung des für einen Tag zu erwartenden Datenvolumens vorgenommen. Die Anzahl der Messwerte pro Minute wurden auf der Grundlage der in 5.3 verwendeten Beispieldatensätze abgeschätzt und bezieht sich auf eine durchschnittliche Wassertiefe von 4000 m. Der Speicherbedarf pro Datenzeile wurde

auf der Grundlage der in Abbildung 5.1 dargestellten Datenbanktabellen ermittelt, wobei für den von ArcSDE verwalteten Shape-Typ 'Point' (s. ArcSDE Datenverwaltung, S. 33) jeweils mit einer Größe von 30 Byte gerechnet wurde.

Tabelle 5.1: Abschätzung des Datenvolumens der Tiefenmesssysteme pro Tag

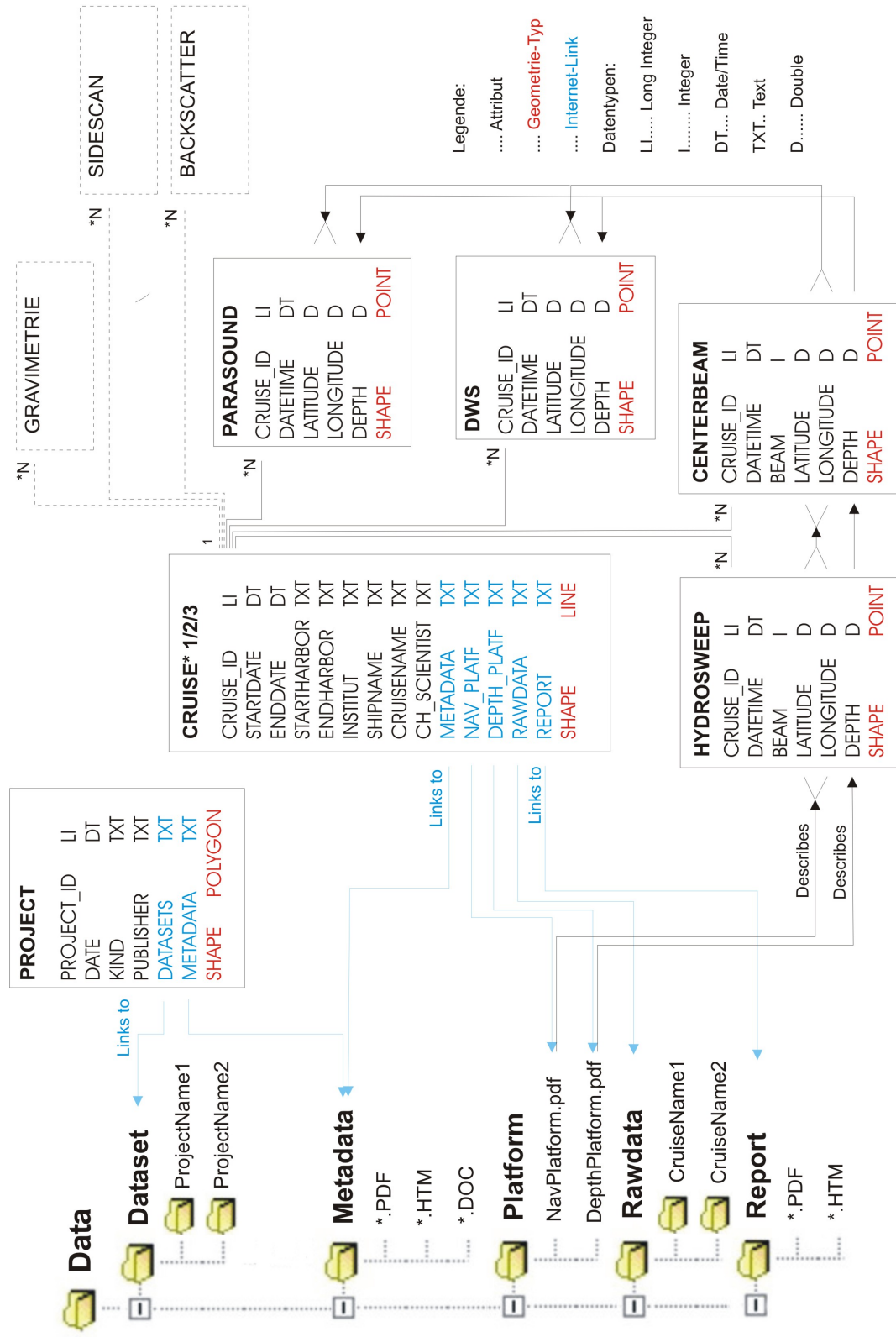
Messsystem	Messwerte pro Minute	Messungen pro Tag	Speicherbedarf pro Datenzeile	Speicherbedarf pro Tag
DWS	6	8640	60 Byte	0.52 MB
Parasound	5	7200	60 Byte	0.44 MB
Hydrosweep	240	345600	60 Byte	20 MB

Bei etwa 100 Messtagen pro Jahr ergibt sich somit ein Datenvolumen von 52/44/2000 MB. Aufgrund des hohen Datenvolumens sind die Hydrosweep-Daten daher in diesem Umfang nicht zur Präsentation im ArcIMS Viewer geeignet. Um aber dennoch die Einsatzbereiche von Hydrosweep aufzuzeigen, werden statt der 59 Beams (s. Hydrosweep, S. 7) jeweils nur die Centerbeams¹ visualisiert. Das Datenvolumen der Centerbeams beträgt nach 100 Messtagen etwa 35 MB und liegt damit in der Größenordnung von DWS und Parasound. Alle Tiefendaten eines Messsystems werden jeweils in einer separaten Datenbanktabelle gespeichert und über einen Index mit der 'Cruise'-Tabelle verknüpft. Dadurch ist es zum einen möglich die Tiefendaten unabhängig voneinander im ArcIMS Viewer zu visualisieren und zum anderen wird über den gemeinsamen Index die Datenrecherche erleichtert.

Außerdem soll der Nutzer laut Zielsetzung (3.1) die Möglichkeit haben über die Abfrageergebnisse direkt auf Roh- und Metadaten zuzugreifen. Da diese aber aufgrund ihrer Beschaffenheit auf einem Fileserver abgelegt sind, kann die Verknüpfung nicht über einen Index realisiert werden. Stattdessen wird die Verbindung zu den Roh- und Metadaten über einen Internet-Link hergestellt. Der Link kann allerdings nicht direkt in der Metadaten-tabelle verwaltet werden, sondern wird erst durch den ArcIMS Viewer bei der Anzeige der Abfrageergebnisse dem entsprechenden Attribut zugewiesen (s. 5.6 Projektorientierte Anpassung). Je nach Format der verknüpften Daten werden diese entweder im Browser angezeigt (HTML, XML, PDF) oder zum Herunterladen bereitgestellt (SURF, DUX). Für den Zugriff auf Projektdaten wird eine weitere Metadaten-tabelle ('Project') in der Datenbank angelegt, die u.a. Informationen zum Bearbeiter und zur Art (Karte, Diplomarbeit, Dissertation) der durchgeführten Projekte enthält. Der räumliche Bezug wird dabei über Polygone, die die Grenzen der zu Grunde liegenden Bearbeitungsgebiete markieren,

¹entspricht der Tiefe, die genau unter der Schiffslängsachse ermittelt wurde

KAPITEL 5. PRAKTISCHE UMSETZUNG



*Die 'Cruise'-Tabelle existiert insgesamt dreimal mit den gleichen Attributen, aber mit jeweils unterschiedlicher Auflösung des zugrundeliegenden Fahrtverlaufes (s. 5.3.2 Vorbereitung)

Abbildung 5.1: Datenmodell bestehend aus Datenbank und Fileserver.

hergestellt. Die Endprodukte eines jeden Projektes sind ebenfalls wieder über Internet-Links mit den entsprechenden Attributen der Metadatentabelle verknüpft.

Theoretisch wäre auch eine Verknüpfung der 'Project'- mit der 'Cruise'-Tabelle in der Datenbank möglich, um z.B. Abfragen der Art „Auf welche Daten welcher Expeditionen wurde innerhalb eines Projektes zugegriffen?“ zu ermöglichen. Allerdings würde es sich hierbei um eine m:n-Beziehung handeln, was zur Folge hätte, dass eine zusätzliche 'Join'-Tabelle anzulegen wäre. Da aber Abfragen im ArcIMS Viewer über mehrere Tabellen nicht durchgeführt werden können und die Ergebnisse dieser Abfragen bereits in den Metadaten enthalten sind, wird eine solche Verknüpfung in der vorliegenden Arbeit nicht umgesetzt.

Die Kombination aus Datenbank und Fileserver ermöglicht es, dass alle raumbezogenen Daten in einem einheitlichen Ablagesystem organisiert werden und somit für die direkte und gemeinsame Weiterverarbeitung (z.B. mit ArcMap) zur Verfügung stehen. Weiterhin können Daten anderer Fachbereiche, wie in Abbildung 5.1 angedeutet, ebenso problemlos ergänzt werden, wie Daten anderer Organisationen, da die über Internet-Links verknüpften Dateien auf dem Fileserver beliebig anpassbar sind.

5.2 Aufbau der Datenbank

Das konzipierte Datenmodell wird mit Hilfe der ArcCatalog-Funktionen (s. ArcCatalog, S. 31) in eine *Geodatabase* (s. Geodatabase, S. 32) umgesetzt. Grundsätzlich stehen dazu zwei verschiedene Typen zur Verfügung: die *Personal*- und die *ArcSDE-Geodatabase* (ESRI 2002b, S. 10). Da aber die *Personal Geodatabase* nur das Anlegen einer lokalen Datenbank (Microsoft Access) erlaubt und daher den gleichzeitigen Zugriff mehrerer Nutzer nicht unterstützt sowie keine Möglichkeiten bietet, Rasterdaten zu verwalten (s. Abb. A.2), kommt für den Aufbau der Datenbank nur die *ArcSDE Geodatabase* in Frage. Diese befindet sich physikalisch gesehen auf dem SQL Server 2000.

Aufgrund dessen, dass nur über den ArcSDE Application Server auf die Funktionalitäten des SQL Servers zugegriffen werden kann (s. 4.2 Microsoft SQL Server 2000), muss zunächst ein entsprechender ArcSDE Service gestartet werden (ESRI 2002d, S. 39). Dies wird im ArcCatalog mit dem Anlegen einer räumlichen Datenbankverbindung zum SQL Server realisiert. In dem in Abbildung 5.2 dargestellten Dialogfenster sind die dafür notwendigen Parameter eingetragen. Auf der Grundlage dieser räumlichen Datenbankverbindung kann das konzipierte Datenmodell umgesetzt werden.

Alle raumbezogene Daten werden innerhalb einer *Geodatabase* in *Feature Classes* verwaltet (s. ArcSDE Datenverwaltung, S. 33). Für jede Tabelle aus dem Datenmodell (s. Abb. 5.1) wird somit eine separate *Feature Class* mit den entsprechend angegebenen Eigenschaften erstellt. Zusätzlich können die *Feature Classes* in einem *Feature Dataset* („Merkmalsraum“)

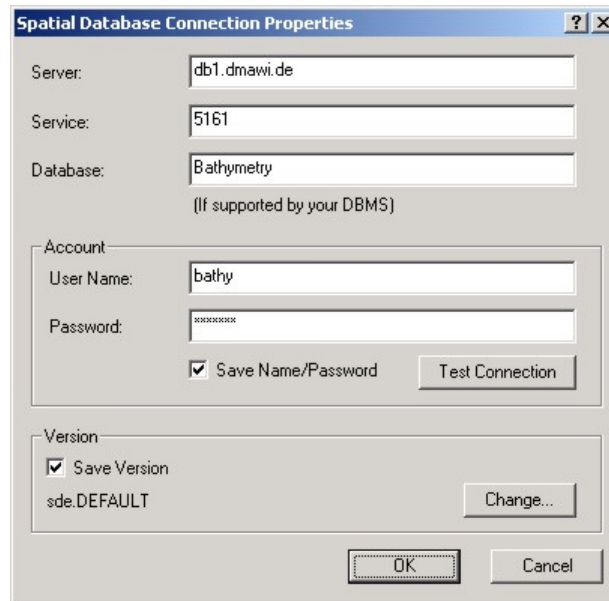


Abbildung 5.2: Einrichten einer räumlichen Datenbankverbindung

organisiert werden. Dies hat den Vorteil, dass für alle in den *Feature Classes* enthaltenen Datensätze sowohl ein einheitlicher räumlicher Bezug als auch ein gemeinsamer Wertebereich für die X,Y,Z-Koordinaten definiert werden kann.

In der vorliegenden Arbeit wurde ein *Feature Dataset* 'Antarktis' mit einer Polar Stereographischen Projektion erstellt, wobei der Zentral-Meridian bei 0° und der Standardparallele Breitenkreis bei -65° liegt. Die Wertebereiche für die Koordinaten wurden entsprechend der Größe des zu bearbeitenden Gebietes angepasst (s. Abb. 5.3). Die 'Precision' gibt dabei jeweils die Anzahl der internen Systemeinheiten je Messeinheit an, d.h. bei einem Wert von 1000 werden die Koordinaten auf drei Nachkommastellen gespeichert².

Innerhalb des *Feature Datasets* wurden danach die verschiedenen *Feature Classes* angelegt. Dabei war vor allem auf die Auswahl des entsprechenden 'Geometry Types' und auf die korrekte Definition der Zahlentypen (s. Abb. 5.4) zu achten. Die 'Precision' gibt dabei die Anzahl der signifikanten Ziffern an, während 'Scale' die Anzahl der Dezimalstellen festlegt (ESRI 2002b, S. 19 ff.). Die Felder 'Latitude', 'Longitude' und 'Depth' sind lediglich Attribute der *Feature Class* 'Centerbeam'. Sie haben somit keinen Einfluss auf die Lage des Punktes und sind daher auch nicht an die entsprechenden Wertebereiche und an den räumlichen Bezug des *Feature Datasets* gebunden. Die für die Lage eines geographischen

²Nach der Auswahl der Wertebereiche wird immer die maximal mögliche 'Precision' berechnet und eingetragen. Sie entspricht damit nicht unbedingt den realen messtechnischen Möglichkeiten.

5.2. AUFBAU DER DATENBANK

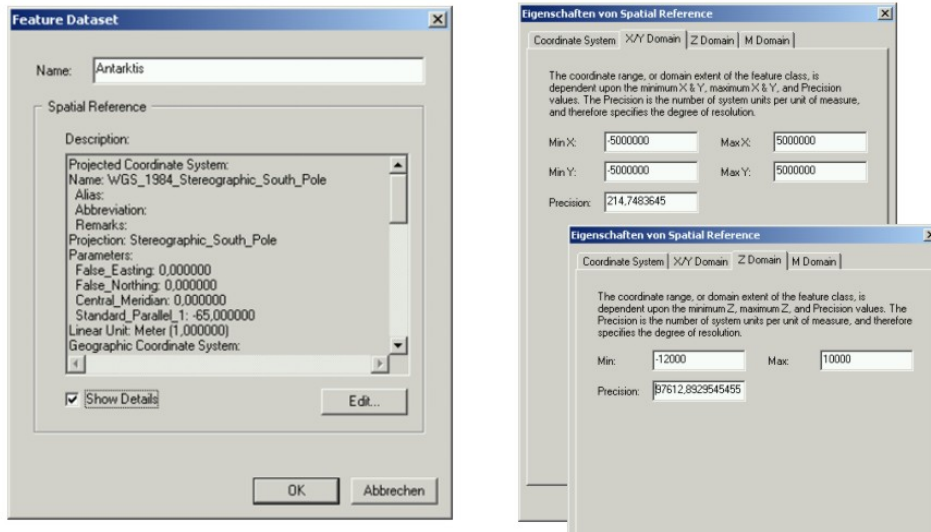


Abbildung 5.3: Erforderliche Einstellungen (Projektion und Wertebereiche) beim Anlegen des *Feature Datasets* 'Antarktis'

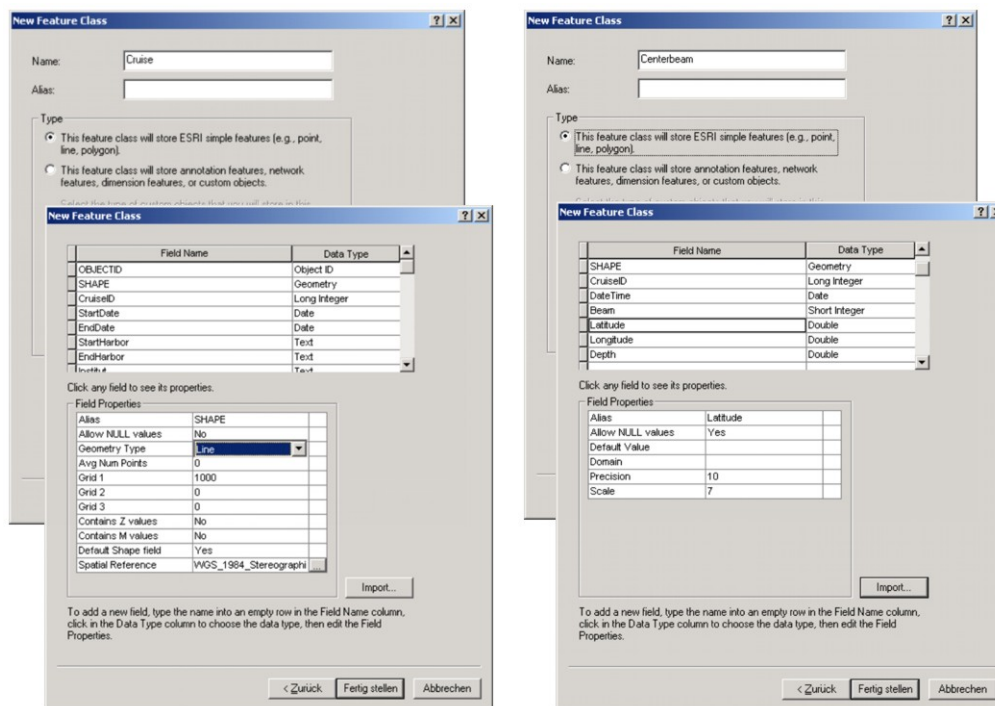


Abbildung 5.4: Auswahl des 'Geometrie Typs' sowie Definition der Zahlendatentypen beim Anlegen einer *Feature Class*

Objektes (Punkt, Linie, Fläche) entscheidenden Koordinaten werden in dem jeweiligen 'Geometry Typ' verwaltet (s. ArcSDE Datenverwaltung, S.33).

5.3 Einlesen von Beispieldatensätzen in die Datenbank

Anhand von Beispieldatensätzen aus dem Bereich des Südozeans (50° - 90° südlicher Breite) wird im ersten Abschnitt gezeigt, welche Formatierungen und Konvertierungen der Tiefendaten notwendig sind, bevor diese in die *Geodatabase* eingelesen werden können. Der zweite Teil beschäftigt sich mit dem Import der Daten und mit der Bearbeitung und Anpassung der beiden Metadatentabellen.

Als Grundlage dienen die auf den beiden Antarktis-Expeditionen ANT-XVIII (2000/01) und ANT-XIX (2001/02) aufgezeichneten Daten. Im Mittelpunkt stehen dabei die Fahrtabschnitte ANT-XVIII/5a, ANT-XIX/2 und ANT-XIX/5, da dort jeweils die drei Tiefenmesssysteme DWS, Parasound und Hydrosweep gleichzeitig (parallel) in Betrieb waren. Sowohl die entsprechenden Tiefendaten für DWS und Parasound als auch die Positionsdaten zur Erstellung des linienförmigen Fahrverlaufes (*Trackline*) für die 'Cruise'-Tabelle wurden aus der PODAS-Datenbank extrahiert (s. 2.2.1 PODAS). Die Hydrosweep- bzw. Centerbeam-Daten wurden dagegen, aufgrund der in der Hydrosweep-Auswertung vorgenommenen Navigationseditierung (s. 2.1.3 Datenfluss/ Formate), den HYD-Dateien entnommen. Bei der Zusammenstellung der notwendigen Informationen für die Metadatentabellen ('Cruise' und 'Project') konnte auf verschiedene AWI-interne Datenquellen zurückgegriffen werden (u.a. Expeditions-Datenbank, Berichte zur Polarforschung). Für den räumlichen Bezug der 'Project'-Tabelle wurden die vorhandenen Umringspolygone der BCWS-Karten (Bathymetric Charts of the Weddell Sea) genutzt.

5.3.1 Vorbereitung

Die Tiefen- und Positionsdaten wurden aus der PODAS-Datenbank über die Eingabe der jeweiligen Zeiträume der einzelnen Fahrtabschnitte extrahiert. Für die Auswahl der notwendigen Parameter sowie für die Definition des Ausgabeformates (s. 2.2.1 PODAS) wurde eine bereits bestehende Konfigurationsmaske verwendet. Nach der Extraktion stand im Ergebnis für jeden Fahrtabschnitt eine DAT-Datei zur Verfügung, die in sekundlicher Auflösung die gewünschten Information enthielt. Diese konnte mit dem vorhandenen Programm 'podas2nak' in das NAK-Format konvertiert werden (s. Abb. 5.5). Zur einfacheren Handhabung der Datenmengen wurde dabei für jeden Messtag eines Fahrtabschnitts eine separate Datei erzeugt. Auf der Grundlage dieser Dateien erfolgte die Datenzusammenstellung für

5.3. EINLESEN VON BEISPIELDATENSÄTZEN IN DIE DATENBANK

die 'Parasound'- und 'DWS'- sowie der 'Cruise'-Tabelle (s. Abb. 5.1).

DATUM	ZEIT	NAV-SENSOR	LAT-MINS	LON-MINS	LAT-NACOS	LON-NACOS	DWS	PARA-SOUND	CENTER-BEAM	GYRO	DOLOG	
09-03-01	20:07:49	9	-69.23954	-107.98186	-69.23952	-107.98243	0.	4069.	4084.	276.5	8.5	0.1
09-03-01	20:07:50	9	-69.23954	-107.98198	-69.23952	-107.98245	0.	4069.	4084.	276.4	8.6	0.1
09-03-01	20:07:51	9	-69.23954	-107.98209	-69.23952	-107.98265	0.	4069.	4084.	276.4	8.6	0.1
09-03-01	20:07:52	8	-69.23949	-107.98220	0.00000	0.00000	0.	4069.	4084.	276.4	8.6	0.0
09-03-01	20:07:53	8	-69.23949	-107.98226	0.00000	0.00000	4070.	4069.	4084.	276.4	8.6	0.0
09-03-01	20:07:54	9	-69.23949	-107.98238	-69.23952	-107.98283	0.	4069.	4084.	276.6	8.6	0.0
09-03-01	20:07:55	9	-69.23949	-107.98249	-69.23952	-107.98302	0.	4069.	4084.	276.8	8.5	0.0
09-03-01	20:07:56	9	-69.23949	-107.98260	-69.23952	-107.98320	0.	4069.	4084.	276.8	8.5	0.0
09-03-01	20:07:57	9	-69.23949	-107.98278	-69.23952	-107.98323	0.	4069.	4084.	276.5	8.6	0.1
09-03-01	20:07:58	8	-69.23949	-107.98289	0.00000	0.00000	4071.	4069.	4084.	276.3	8.6	0.1
09-03-01	20:07:59	8	-69.23949	-107.98301	0.00000	0.00000	0.	4069.	4078.	276.2	8.6	0.1
09-03-01	20:08:00	8	-69.23949	-107.98312	0.00000	0.00000	0.	4069.	4078.	276.4	8.5	0.0
09-03-01	20:08:01	9	-69.23949	-107.98324	-69.23952	-107.98372	0.	4069.	4078.	276.7	8.5	0.0
09-03-01	20:08:02	9	-69.23949	-107.98330	-69.23952	-107.98373	0.	4069.	4078.	276.8	8.5	0.0
09-03-01	20:08:03	9	-69.23949	-107.98341	-69.23952	-107.98392	4075.	4069.	4078.	276.7	8.5	0.0
09-03-01	20:08:04	8	-69.23943	-107.98352	0.00000	0.00000	0.	4069.	4078.	276.5	8.5	0.0

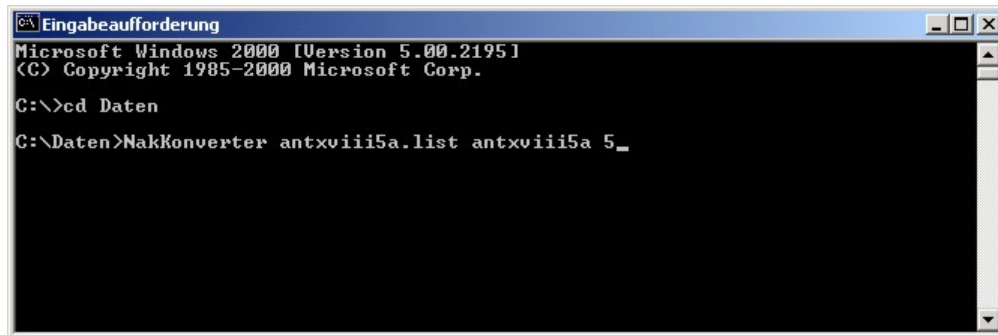
Abbildung 5.5: Ausschnitt aus einer NAK-Datei

Dazu wurde im Rahmen der vorliegenden Arbeit das Programm 'NakKonverter' erstellt. Es extrahiert aus den NAK-Dateien (*.nak) die Tiefendaten für Parasound und DWS und schreibt diese mit den dazugehörigen Attributen in zwei voneinander unabhängige Textdateien (*.txt). Dabei werden nur die Daten ausgelesen, die sich im betreffenden Gebiet (südlich 50° Süd) befinden und die die Vorgaben des Plausibilitätstest erfüllen (s. Anhang B.1). Außerdem werden die in den NAK-Dateien angegebenen „positiven“ Tiefen durch Umkehrung des Vorzeichens in „wirkliche“ Tiefen umgewandelt. Zusätzlich wird vom 'NakKonverter' eine Datei im Generate-Format (*.gen) erstellt, die die Punktinformationen für drei Linien unterschiedlicher Auflösungen enthält, die jeweils den Fahrtverlauf des jeweiligen Expeditionsabschnittes darstellen. Damit besteht die Möglichkeit den Fahrtverlauf in Abhängigkeit des Maßstabs im ArcIMS Viewer zu visualisieren, wodurch die Übersichtlichkeit im entsprechenden Ansichtsfenster gesteigert und die benötigte Aufbauzeit verkürzt wird.

Das Programm selbst wird in einer Windows-Konsole in demselben Verzeichnis aufgerufen, in dem sich auch die NAK-Dateien befinden. Dabei sind die folgenden vier Parameter anzugeben: Programmname, Name der Listing-Datei³, Name des Fahrtabschnittes sowie

³Die Listing-Datei enthält die Namen aller NAK-Dateien eines Fahrtabschnittes. Dabei steht jeweils pro Zeile ein Dateiname.

der Wert der CruiseID⁴. Ein Beispiel für den Aufruf des Programms zeigt Abb. 5.6. Mit der Ausführung von 'NakKonverter' sind die Vorbereitungen für die Parasound- und DWS-Daten abgeschlossen.



```
Eingabeaufforderung
Microsoft Windows 2000 [Version 5.00.2195]
Copyright 1985-2000 Microsoft Corp.
C:\>cd Daten
C:\Daten>NakKonverter antxviii5a.list antxviii5a 5_
```

Abbildung 5.6: Aufruf des Programms 'NakKonverter' mit den erforderlichen Parametern in einer Windows-Konsole.

Die Hydrosweep- bzw. Centerbeam-Daten wurden aus den in der Auswertung erzeugten HYD-Dateien (*.hyd) ausgelesen (s. 2.1.3 Datenfluss/Datenformate). Dazu erfolgte zunächst unter UNIX der Aufruf des vorhandenen Programms 'refoBathy'. Dieses erstellte für jeden Messtag eine Hydrosweep- und eine Centerbeam-Datei im ASCII-Format. Nach dem Übertragen der ASCII-Dateien nach Windows, wurde das neuerstellte Programm 'Ascii-Konverter' aufgerufen, welches die Lage der Tiefendaten hinsichtlich des betreffenden Gebietes prüft und alle gültigen Werte in eine neue Textdatei (*.txt) schreibt (s. Anhang B.2). Der 'AsciiKonverter' ist ebenfalls eine Konsolen-Anwendung und wird mit den gleichen Parametern aufgerufen wie der NakKonverter (s. Abb. 5.6).

5.3.2 Einlesen

Um raumbezogene Daten in eine *Geodatabase* zu importieren, müssen diese zuvor in den abstrakten geometrischen Datentyp „Shape“ (u.a. Punkt, Linie oder Fläche) konvertiert werden. ArcToolbox bietet dazu den 'Generate to Coverage Wizard', der es erlaubt, aus Positionsdaten, die im Generate-Format vorliegen, einen gewünschten „Shape“-Typ zu erzeugen. Diese Funktion wurde im vorliegenden Fall genutzt, um aus den mit dem 'Nakkonverter' erstellten Generate-Dateien die *Tracklines* unterschiedlicher Auflösung zu erzeugen. Im Ergebnis stand für jeden Fahrtabschnitt ein Linien-Coverage⁵ zur Verfügung

⁴Entspricht dem Wert, der dem entsprechenden Fahrtabschnitt in der 'Cruise'-Tabelle zugewiesen wird.

⁵ESRI-internes Datenformat, zur Verwaltung von geographischen Objekten

5.3. EINLESEN VON BEISPIELDATENSÄTZEN IN DIE DATENBANK

(s. Abb. 5.7). Dabei konnte es durchaus sein, dass für eine Auflösungsstufe mehrere Linien erzeugt wurden, da ArcInfo jeweils nur maximal 500 Punkte je Linie verwalten kann.

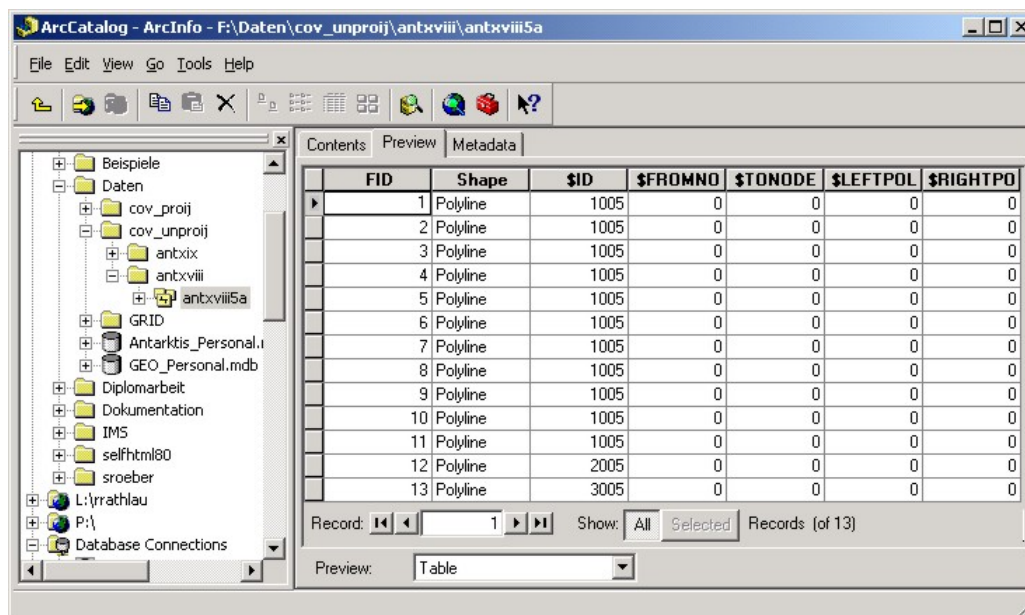


Abbildung 5.7: Linien-Coverage des Fahrtabschnitts ANT-XVIII/5a im Preview des ArcCatalogs. Jede Zeile enthält eine Linie. Das Feld '\$ID' gibt dabei an, zu welcher Auflösung die jeweilige Linie gehört.

Nachdem den Linien-Coverages mit dem 'Define Projection Wizard' der ArcToolbox ein geographisches Referenzsystem zugewiesen wurde, konnten sie mit dem 'Project Wizard' in das Zielsystem projiziert werden. Um den Linien-Coverages die Attributwerte zuweisen zu können, wurden sie in eine *Feature Class* eingelesen, die zusätzlich zu den Attributen der 'Cruise'-Tabelle ein Feld 'CoverageID' vom Datentyp Long Integer enthält. Diesem Feld wurde beim Import die '\$ID' des Linien-Coverages übergeben. Danach konnten mit Hilfe der Funktionalitäten von ArcMap die entsprechenden Attributwerte eingetragen werden. Abschließend erfolgte der Import der *Feature Class* in die 'Cruise'-Tabellen der *ArcSDE Geodatabase*. Anhand der 'CoverageID' konnten dabei die Linien einer Auflösungsstufe jeweils der richtigen 'Cruise'-Tabelle zugeordnet werden (s. Abb 5.1).

Die Umringspolygone der BCWS-Karten lagen bereits in der Polar Stereographischen Projektion vor. Sie konnten daher direkt in die 'Project'-Tabelle der *ArcSDE Geodatabase* eingelesen werden. Um aber in ArcMap die entsprechenden Attributwerte ergänzen zu können, war es zuvor notwendig, im ArcCatalog den Befehl 'Register as Versioned' im Kontext-Menü des *Feature Datasets* 'Antarktis' aufzurufen. Mit diesem Befehl wird erreicht, dass ein Nutzer Veränderungen am Datenbestand einer *Feature Class* vornehmen

kann, wobei der gleichzeitige Zugriff auf diese durch andere Nutzer gewährleistet bleibt (ESRI 2002b, S. 293 ff). Nach der Attributzuweisung wurden die vorgenommenen Veränderungen mit dem Befehl 'Compress Database', aus dem Kontext-Menü der räumlichen Datenbankverbindung (s. 5.2 Aufbau der Datenbank), endgültig in die 'Project'-Tabelle der *ArcSDE Geodatabase* übernommen.

Für den Import der Tiefendaten (DWS, Parasound, Hydrosweep/Centerbeam) konnte nicht der 'Generate to Coverage Wizard' der ArcToolbox verwendet werden, da dieser es nicht erlaubt, zusätzliche Attribute (z.B. Datum/Uhrzeit) mit in die *Coverage*-Datei zu übernehmen. Stattdessen wurde die Funktion 'Create Feature Class -> From XY Table' des ArcCatalogs genutzt. Allerdings war es daher zunächst notwendig die Tiefendaten in eine Tabellenstruktur zu überführen. Dazu wurden die jeweiligen Textdateien (s. 5.3.1 Vorbereitung) in eine Microsoft Access Datenbank importiert. Für jede eingelesene Spalte wurde dabei der entsprechend im Datenmodell angegebene Datentyp festgelegt (s. Abb. 5.1). Um auf die erstellten Datenbanktabellen zugreifen zu können, wurde im ArcCatalog eine OLE DBVerbindung (Object Linking and Embedding Database) zur Access Datenbank eingerichtet. Damit war es nun möglich, die oben genannte Funktion im Kontext-Menü einer Datenbanktabelle im ArcCatalog aufzurufen (s. Abb. 5.8).

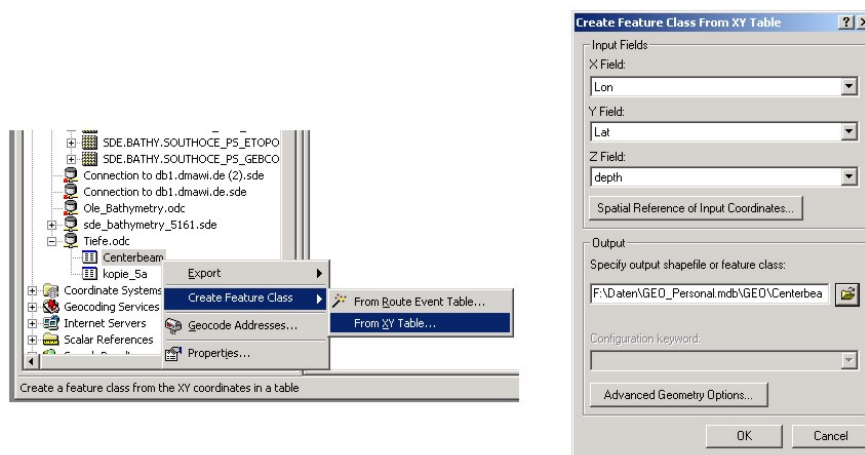


Abbildung 5.8: Aufruf der Funktion 'Create Feature Class -> From XY Table' im ArcCatalog.

In dem angezeigten Dialogfenster waren noch folgende Einstellungen vorzunehmen: Auswahl der 'Tiefe' als Z-Koordinate, Definition eines geographischen Referenzsystems sowie die Angabe des Pfadnamens für die neue *Feature Class*⁶. Im Ergebnis stand somit jeweils

⁶Bei der Verwendung eines *Shapefiles* anstelle einer *Feature Class* wurden nur die Datumsangaben übernommen. Dies ist aber für den Vergleich der Tiefendaten untereinander nicht ausreichend.

für Parasound, DWS, Hydrosweep und Centerbeam eine *Feature Class* mit den erforderlichen Attributen zur Verfügung. Diese wurden mit dem 'Project Wizard' der ArcToolbox in das Zielsystem projiziert und konnten anschließend in die entsprechenden *Feature Classes* der *ArcSDE Geodatabase* importiert werden.

5.4 Erstellen von standard-konformen Metadaten

Auf der Grundlage der beiden Metadatentabellen 'Project' und 'Cruise' können im ArcIMS Viewer inhaltliche Abfragen durchgeführt werden, mit dem Ziel, vorhandene Datensätze in dem Datenbanksystem ausfindig zu machen. Um aber dem Nutzer die Möglichkeit zu geben, die gefunden Datensätze z.B. auf Anwendbarkeit zu untersuchen (s. Wozu Metadaten?, S. 20), müssen detailliertere Informationen verfügbar sein. Dazu ist es notwendig zusätzliche Metadatendateien anzulegen, auf die über den jeweils angegebenen Link zugegriffen werden kann (s. Abb. 5.1). Sowohl für die Projekt- als auch für die Tiefendaten wurde dies im Rahmen der vorliegenden Arbeit mit Hilfe der Funktionalitäten des ArcCatalogs realisiert.

Der ArcCatalog bietet mit dem Metadateneditor die Möglichkeit für jedes *Feature Dataset* und für jede *Feature Class* eine Metadatenfile anzulegen, die direkt mit in der entsprechenden *Geodatabase* verwaltet wird. Da aber im vorliegenden Fall sowohl die 'Project'- als auch die 'Cruise'-Tabelle viele verschiedenartige Datensätze enthalten können, ist eine umfassende und detaillierte Beschreibung durch jeweils eine Metadatenfile nicht gegeben. Um dennoch ausführliche Informationen zu den einzelnen Datensätzen zur Verfügung stellen zu können, wurde ein weiteres *Feature Dataset* namens 'Metadata' in der *ArcSDE Geodatabase* angelegt. In diesem wurde für jeden Datensatz der beiden Metadatentabellen eine *Feature Class* mit einem eindeutigen Namen erzeugt. Der dazu notwendige räumliche Bezug wurde entsprechend über das zugrundeliegende geographische Objekt (Linie oder Fläche) hergestellt.

Mit dieser Anordnung war es möglich, jeden vorhandenen Datensatz mit einer eigenen Metadatenfile detailliert zu beschreiben. Zur Erstellung von Metadaten bietet der ArcCatalog zwei verschiedene Editoren, die jeweils auf einen Metadaten-Standard aufsetzen (s. Standards, S. 22): den ISO Wizard (ISO Norm 19115) und den FGDC Editor (CSDGM 2). Da beide Standards inhaltlich die gleichen Informationen verarbeiten können, diese aber untereinander nicht austauschbar sind, wurde die Entscheidung, welcher Standard zum Einsatz kommen sollte, auf der Grundlage der Funktionalität des entsprechend dazugehörigen Editors getroffen. Die Wahl fiel dabei auf den FGDC Editor, da dieser das Bearbeiten von Metadaten, aufgrund der übersichtlichen Struktur, erleichtert.

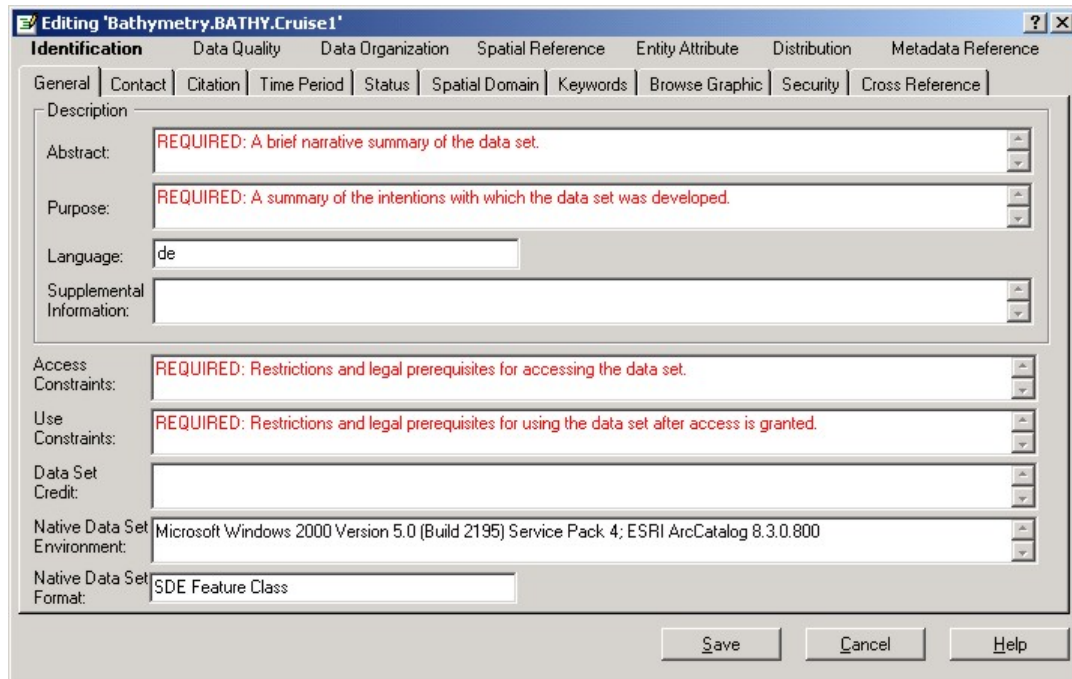


Abbildung 5.9: Der FGDC Metadaten-Editor.

Bevor der Editor mit dem Icon 'edit metadata' in der Toolbar 'Metadata' geöffnet werden kann, muss zunächst die entsprechende *Feature Class* markiert und das Registerblatt 'Metadata' ausgewählt werden. Nach dem Aufruf erscheint das in Abbildung 5.9 dargestellte Dialogfenster. Die Rot hervorgehobenen Texte zeigen dabei an, welche Felder ausgefüllt werden müssen, um die Mindestanforderungen des CSDGM (Content Standard for Digital Geospatial Metadata) zu erfüllen. Darüber hinaus stehen aber auch noch eine Vielzahl von weiteren Eingabefeldern zur Verfügung, die es ermöglichen, den zugrundeliegenden Datensatz umfassend zu beschreiben⁷. Bezüglich der Tiefendaten sei dabei besonders auf den Schwerpunkt 'Data Quality' hingewiesen.

Nach der Eintragung aller erforderlichen und notwendigen Informationen wurde der Metadatenatz gespeichert, in eine HTML-Datei exportiert und auf dem Fileserver im Ordner 'Metadata' abgelegt⁸. Die abschließende Verknüpfung über einen Link zur 'Project'- oder 'Cruise'-Tabelle erfolgte im ArcIMS Viewer (s. 5.6 Projektorientierte Anpassung).

Aus der Verwaltung der Metadaten mit ArcCatalog ergeben sich die beiden folgenden wesentlichen Vorteile: Zum einen können mit Hilfe der Suchmaschine des ArcCatalogs die

⁷Eine ausführliche Anleitung zur Bedienung des FGDC Editors kann ESRI 2001b (S. 129 ff) entnommen werden.

⁸Ein vollständiger Metadatenatz ist im Anhang C abgedruckt.

Metadaten-dateien nach inhaltlichen, zeitlichen und geographischen Informationen durchsucht werden. Im Ergebnis werden dabei nicht die zutreffenden Metadaten-dateien selbst, sondern die jeweils zugrundeliegenden Datensätze (z.B. *Feature Classes*) zurückgegeben. Zum anderen ist es durch die Verwendung des FGDC Editors, der die Einhaltung des Metadaten-Standards gewährleistet, möglich, die entsprechende Metadaten-datei jederzeit problemlos zu bearbeiten und zu aktualisieren und diese auch innerhalb eines Metadaten Services zu publizieren. Damit kann die Metadaten-datei dem Informationsaustausch über die sogenannten „Catalog Services“ zur Verfügung gestellt werden (s. Was sind Metadaten?, S. 19).

5.5 Konzeption und Anlage einer ArcIMS Webseite

Um eine graphische Oberfläche zu erstellen, auf deren Grundlage der Nutzer über Internet/Intranet-Anwendungen den vorhandenen bathymetrischen Datenbestand untersuchen kann, ist die Bearbeitung der folgenden drei Schwerpunkte notwendig: Zusammenstellung des zu visualisierenden Karteninhaltes, Einrichtung eines ArcIMS Services sowie Anlage der Webseite. Für die Umsetzung wurden die Werkzeuge des ArcIMS Managers verwendet (s. ArcIMS Manager, S. 37).

Die Zusammenstellung des Karteninhaltes erfolgte mit dem ArcIMS Author. Dazu waren zunächst alle *Feature Classes* der *ArcSDE Geodatabase* hinzuzuladen, die auch im ArcIMS Viewer visualisiert werden sollten (s. Abb. 5.10). Über das jeweilige Eigenschaftsfenster wurden den *Feature Classes* (nachfolgend mit Layer bezeichnet) geeignete Layer-Namen und anschauliche Symbole mit entsprechender Größe und Farbe zugeordnet. Neben der Beschriftung der 'Cruise'- und 'Project'-Daten mit den Attributen 'Cruisename' bzw. 'Kind' (s. Abb. 5.1) erfolgte auch die Einstellung der metrischen Maßstabseinheiten. Um Übersichtlichkeit und kurze Aufbauzeiten im ArcIMS Viewer für die verschiedenen Zoom-Stufen gleichermaßen zu gewährleisten, war es zudem notwendig, für die einzelnen Layer Maßstabsbereiche festzulegen, in denen diese im ArcIMS Viewer dargestellt werden. Dabei wurden die drei 'Cruise'-Layer entsprechend so organisiert, dass jeweils immer nur einer sichtbar ist, während die Layer der Tiefendaten erst ab einem Maßstab von 1:1000000 im ArcIMS Viewer eingeblendet werden. Um innerhalb der Kartenansicht dem Nutzer die Orientierung zu erleichtern, wurden zusätzliche Gitternetzlinien-Layer angelegt. Nach der Fertigstellung der Kartenansicht wurden die vorgenommenen Einstellungen in einem *Map Configuration File* (*.axl) gespeichert.

Ebenso wie die *Feature Classes* lassen sich in ArcIMS Author auch problemlos Bildformate, die über ein sogenanntes „world file“ verfügen, einlesen (s. Anhang A.3). Allerdings können im ArcIMS Author keine SDE Raster aufgerufen und dargestellt werden. Um diese

KAPITEL 5. PRAKTISCHE UMSETZUNG

aber dennoch in einem ArcIMS Viewer visualisieren zu können, müssen sie nachträglich von Hand in einem *Map Configuration File* ergänzt werden. Die Abbildung 5.11 zeigt dazu ein Beispiel.

Im Rahmen der vorliegenden Arbeit wurden zwei *Map Configuration Files* unterschiedlichen Inhaltes angelegt. Das erste enthält die Daten der *Feature Classes*, während im zweiten ein digitales Geländemodell vom Südozean als SDE Raster gespeichert ist.

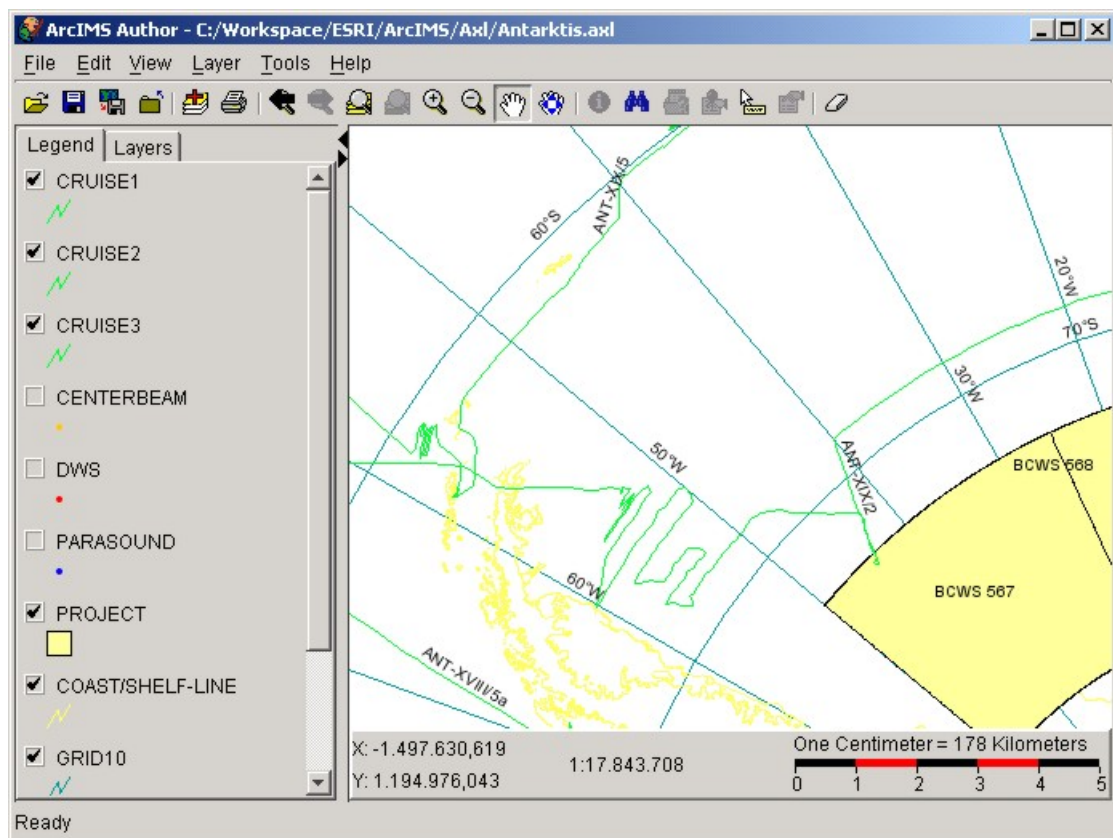
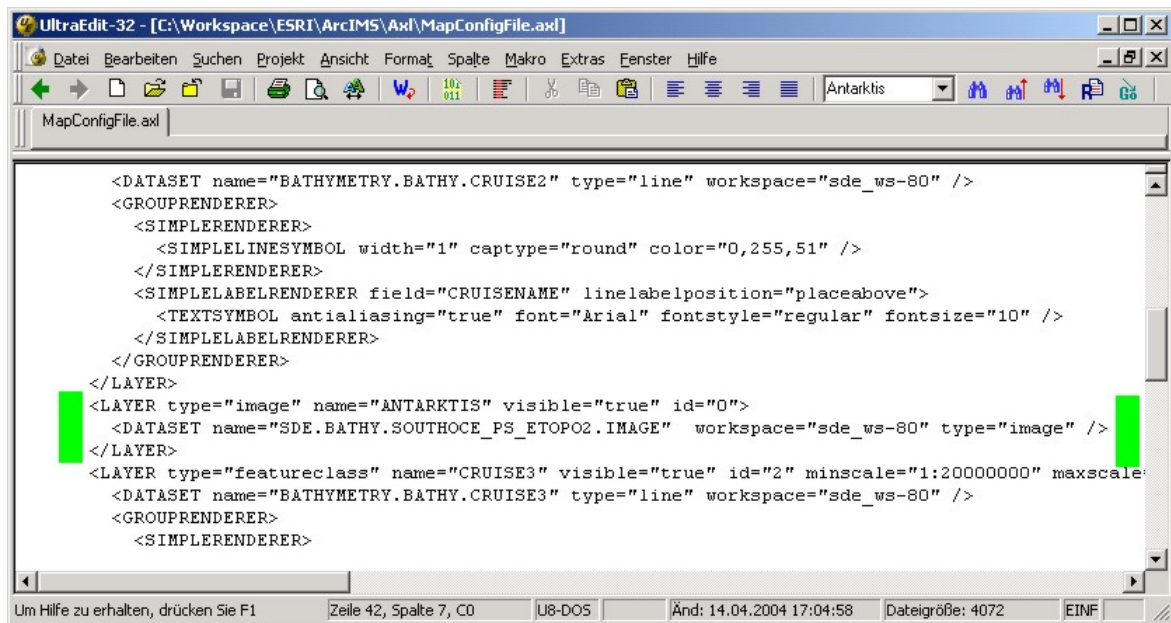


Abbildung 5.10: Die vollständige Kartenansicht im ArcIMS Author

Auf der Grundlage der beiden *Map Configuration Files* erfolgte mit dem ArcIMS Administrator die Einrichtung zwei verschiedener ArcIMS Services. Mit den jeweils zuzuweisenden *Virtual Servern* wurde dabei das Format definiert, in dem die vom Nutzer angeforderten Karten und Daten zusammengestellt werden (s. 4.3.3 ArcIMS Spatial Server). Um einerseits dem Nutzer am Client das größtmögliche Funktionsspektrum zur Verfügung zu stellen und andererseits alle gewünschten Daten zu visualisieren (s. Anhang A.4), wurde für die Darstellung der *Feature Classes* ein *Feature Service* verwendet, während für die Wiedergabe des SDE Rasters die Möglichkeiten des *Image Services* genutzt wurden (s. Abb. 5.12).

5.5. KONZEPTION UND ANLAGE EINER ARCIMs WEBSEITE



```
<DATASET name="BATHYMETRY.BATHY.CRUISE2" type="line" workspace="sde_ws-80" />
<GROUPRENDERER>
  <SIMPLERENDERER>
    <SIMPLELINESYMBOL width="1" captype="round" color="0,255,51" />
  </SIMPLERENDERER>
  <SIMPLELABELRENDERER field="CRUISENAME" linelabelposition="placeabove">
    <TEXTSYMBOL antialiasing="true" font="Arial" fontstyle="regular" fontsize="10" />
  </SIMPLELABELRENDERER>
</GROUPRENDERER>
</LAYER>
<LAYER type="image" name="ANTARKTIS" visible="true" id="0">
  <DATASET name="SDE.BATHY.SOUTHOCOE_PS_ETOPO2.IMAGE" workspace="sde_ws-80" type="image" />
</LAYER>
<LAYER type="featureclass" name="CRUISE3" visible="true" id="2" minscale="1:20000000" maxscale="1:20000000">
  <DATASET name="BATHYMETRY.BATHY.CRUISE3" type="line" workspace="sde_ws-80" />
  <GROUPRENDERER>
    <SIMPLERENDERER>
```

Abbildung 5.11: Dieser Ausschnitt aus einem *Map Configuration File* zeigt die notwendigen Parameter, um ein SDE Raster einzubinden.

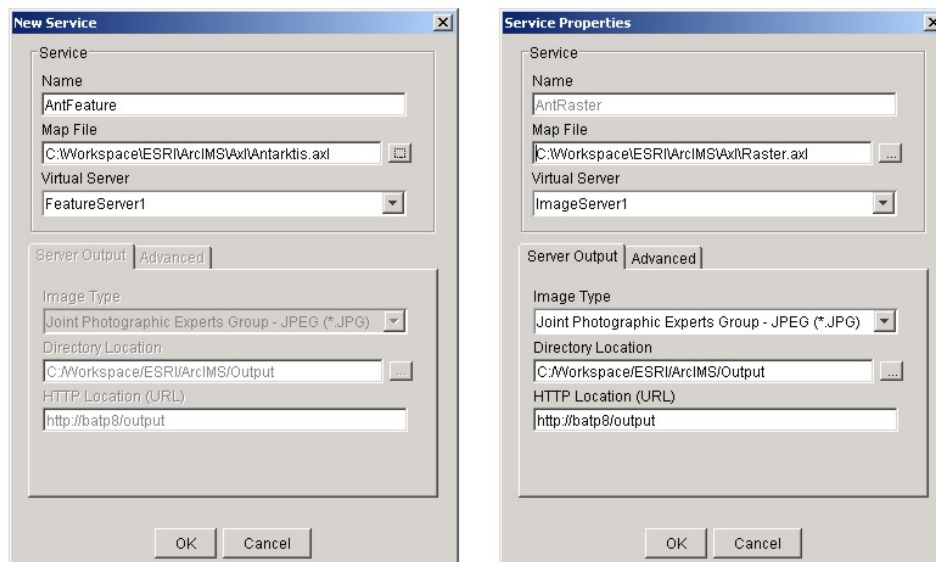
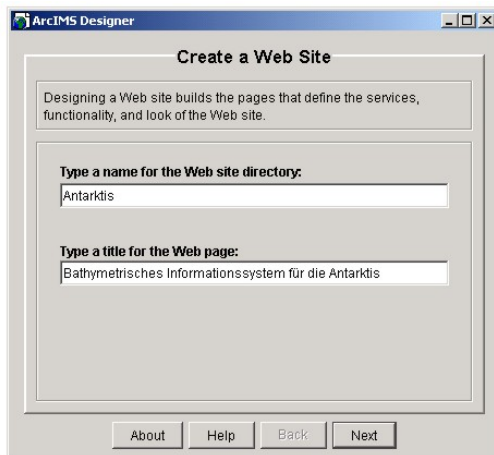


Abbildung 5.12: Einrichten eines *Feature-* und eines *Image Services*

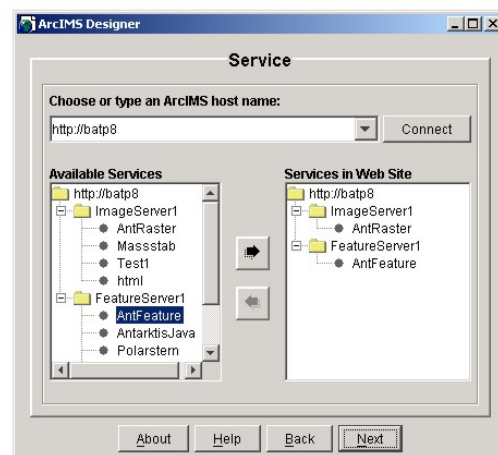
KAPITEL 5. PRAKTISCHE UMSETZUNG

Nach der Einrichtung der beiden ArcIMS Services erfolgte mit Hilfe des ArcIMS Designers die Anlage der Webseite mit entsprechend integriertem ArcIMS Viewer. Dabei besteht grundsätzlich die Möglichkeit zwischen dem HTML-Viewer und den Java-Viewern zu wählen (s. ArcIMS Viewer, S. 38). Der HTML-Viewer konnte aber nicht verwendet werden, da im vorliegenden Fall zwei ArcIMS Services gleichzeitig visualisiert werden sollten. Um nach der Erstellung der Webseite zudem eine projektorientierte Anpassung zu ermöglichen, fiel die endgültige Wahl auf den Java Custom-Viewer.

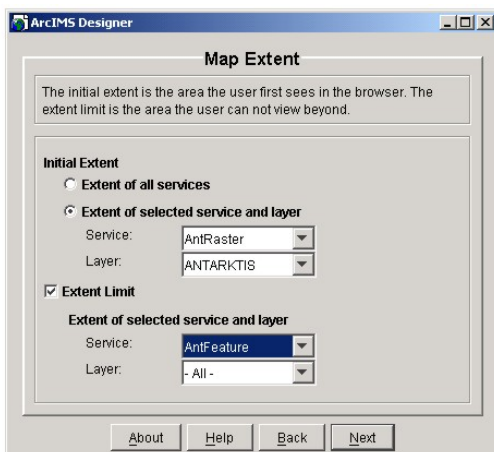
Die folgenden Dialogfenster zeigen wichtige Einstellungen für die Anlage der Webseite:



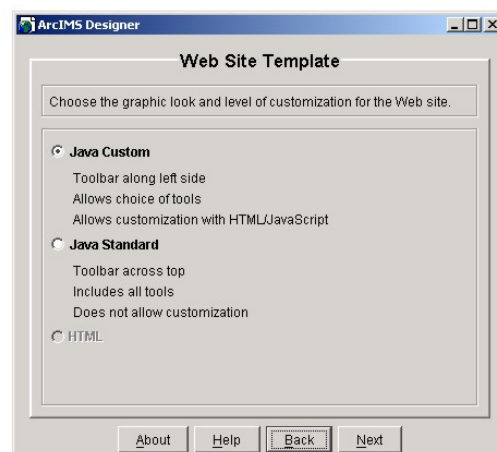
Vergeben von Namen



Auswählen des Image- und Featur-Service

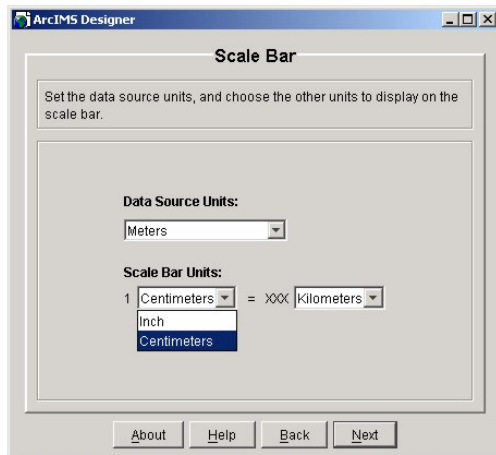


Festlegen der max. Ansichtsbereiche

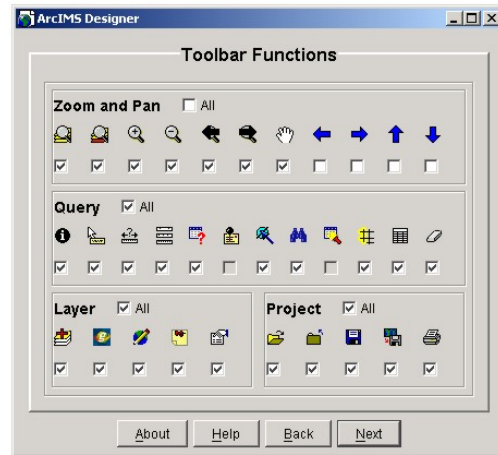


Auswahl des Java Custom-Viewers

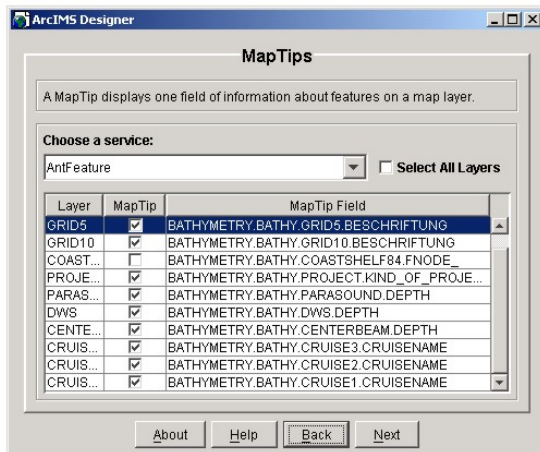
5.5. KONZEPTION UND ANLAGE EINER ARCIMS WEBSEITE



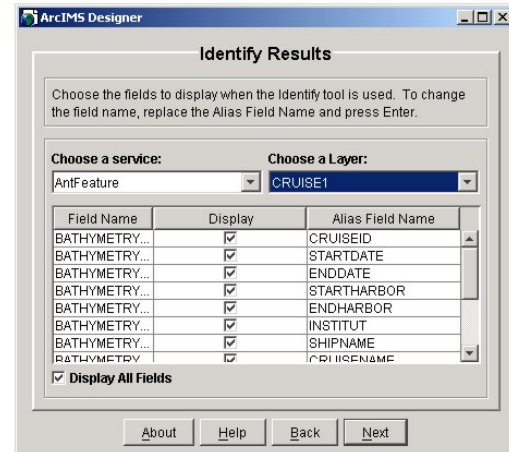
Festlegen der Maßstabseinheiten



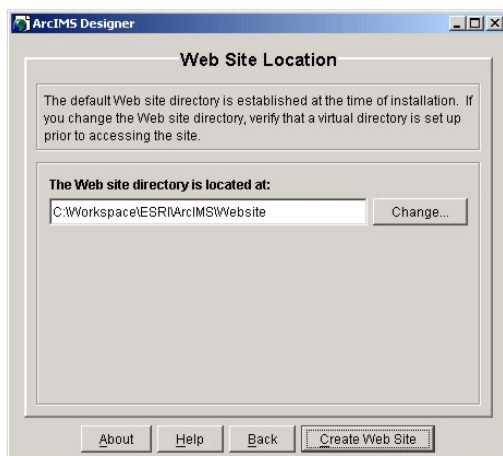
Auswahl der Toolbar-Funktionen



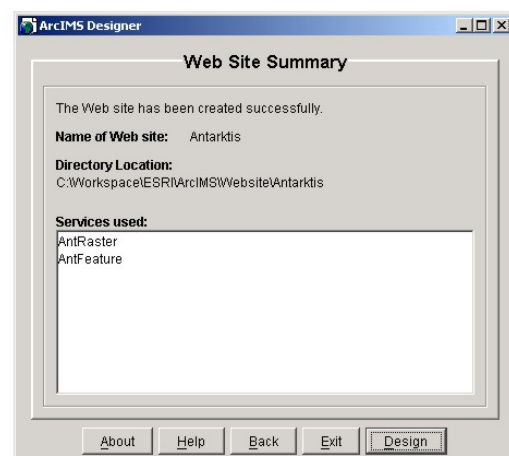
Festlegen der 'Mouse-Over'-Funktion



Vergeben von Alias-Namen



Angabe des Webseitenverzeichnis



Zusammenfassung

KAPITEL 5. PRAKTISCHE UMSETZUNG

Das Ergebnis ist der Ordner 'Antarktis', der alle Dateien enthält, die zur Darstellung der Webseite notwendig sind. Dazu gehören im Hauptverzeichnis die HTML-Dateien (*.htm), die das Layout der Webseite festlegen, die Javascript-Dateien (*.js), die u.a. die Funktionalitäten der Toolbar bestimmen sowie die im Unterverzeichnis 'images' enthaltenen Bilder (*.jpg), die das Aussehen der Werkzeugsymbole definieren. Einen visuellen Eindruck vom Zusammenspiel der einzelnen Komponenten gibt Abbildung 5.13.

Mit Hilfe der in den Javascript-Dateien deklarierten Funktionen konnte der ArcIMS Viewer nachträglich nutzerspezifischen Erfordernissen angepasst werden.

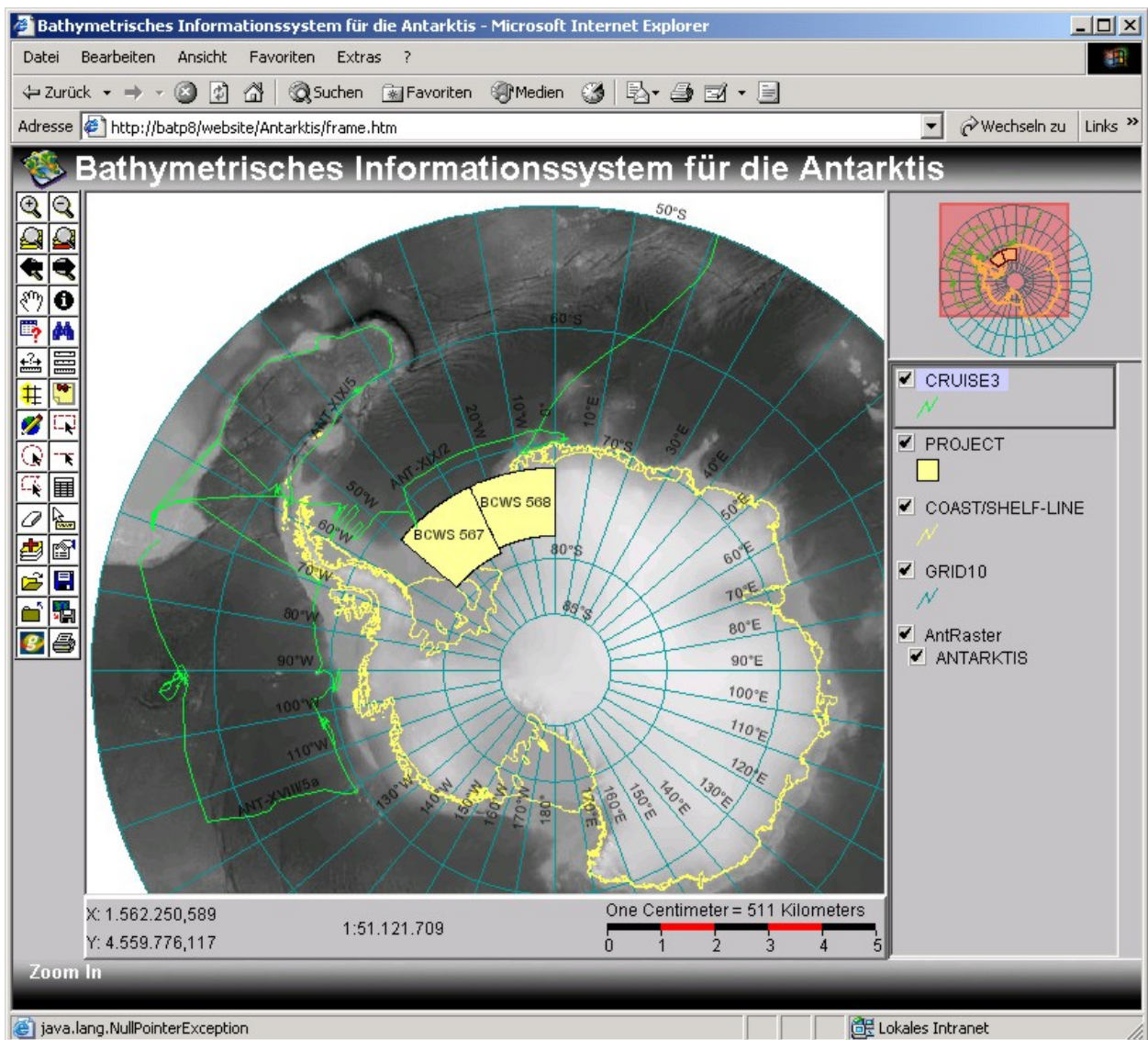


Abbildung 5.13: Aussehen der mit dem ArcIMS Designer erstellten Webseite

5.6 Projektorientierte Anpassung

Um den Nutzer die Möglichkeit zu geben, auf den gesamten bathymetrischen Datenbestand zuzugreifen, müssen die entsprechenden Attribute der Metadatentabellen über einen Internet-Link mit den Verzeichnissen des Fileservers verknüpft werden (s. 5.1 Konzeption des Datenmodells). Dazu sind in den jeweiligen Funktionen der Javascript-Dateien Veränderungen vorzunehmen. Allerdings sind diese in nur sehr begrenztem Maße möglich, da der Java Custom-Viewer zur Visualisierung der Daten und zur Bearbeitung der gestellten Anfragen ein Java2-Applet verwendet. Das hat zur Folge, dass grundsätzlich keine neuen Funktionen hinzugefügt werden können, sondern nur die bereits vorhandenen entsprechend anpassbar sind.

```

if (parent.useShowSelect) {
    document.write('<td align="center" valign="middle">');
    // choose between the following two calls
    // the next two lines will display attributes of selected features
    // in HTML table. . . slower but can be customized easily
    document.write('');
    // the next three lines will display attributes of selected
    // features in Java table. . . faster but cannot be customized
    //document.write('');
    isSecond = !isSecond;
    document.writeln('</td>');
    if (isSecond) document.write('</tr><tr>');
}

```

Abbildung 5.14: Aufruf der Javascript-Funktion 'showSelect()' und Auskommentieren der Applet-Funktion 'displayAttributes()' in der HTML-Datei 'toolbar.htm'

Um bei der Darstellung der Metadatentabellen im Java Custom-Viewer die jeweiligen Attribute mit Internet-Links zu versehen, war es zunächst notwendig, statt der Applet-Funktion 'displayAttributes()' die Javascript-Funktion 'showSelect()' in der HTML-Datei 'toolbar.htm' aufzurufen (s. Abb 5.14). Diese Funktion ermöglicht es, die Ergebnisdatensätze einer vorgenommenen Selektion in einer HTML-Tabelle darzustellen. Beim zeilenweisen Aufbau der HTML-Tabelle konnten somit den jeweiligen Attributen die entsprechenden Internet-Links zugewiesen werden. Dazu war es zuvor erforderlich, die Spaltennamen der

Metadatentabelle aus dem Antwort-'String' des Servers zu extrahieren und sie innerhalb einer 'CASE'-Schleife abzufragen. Auf der Grundlage der Attributwerte wurden danach die Links zusammengesetzt. Die Abbildung 5.15 zeigt dazu zwei verschiedene Link-Typen⁹: In der Spalte 'Metadata' wird auf eine konkrete Datei verwiesen, wohingegen über die Spalte 'Rawdata' der Zugriff auf einen Ordner möglich ist.

```
//Spaltennamen werden abgefragt, ob sie einen Link enthalten sollen
for (var j=1;j < theFields.length; j++) {
  //teilt den Verbindungsnamen z.B.: BATHYMETRY.BATHY.CRUISE.CRUISEID
  //in BATHYMETRY,BATHY,CRUISE,CRUISEID
  //der letzte Wert ist der eigentliche Spaltenname, dieser wird in
  //der 'case'-Schleife abgefragt
  var theSpalte = theFields[j].split(".");
  var numSpalte = theSpalte.length;
  //konkreter Attributwert einer Zeile
  thisValue = theValues[j];
  sendValue = "<td><font face='Arial' size='-3'>";
  switch (theSpalte[numSpalte-1]) {
    case "METADATA":
      sendValue = sendValue + "<a target='_new | _blank'
        href='http://batp8/website/Data/Metadata/" + thisValue + "'>"
        + thisValue + "</a></font></td>";
      break
    case "RAWDATA":
      sendValue = sendValue + "<a target='_new | _blank'
        href='http://batp8/website/Data/Rawdata/" + thisValue + "/">"
        + thisValue + "</a></font></td>";
      break
  }
}
```

Abbildung 5.15: Link-Typen: Verweis auf eine Datei (Metadata) und Verweis auf einen Ordner (Rawdata)

Ein weiterer Vorteil bei der Verwendung der HTML-Tabelle besteht darin, dass Ergebnisdatensätze gleicher Attributwerte, die aufgrund der begrenzten Stützpunktanzahl einer Linie möglich sind (s. 5.3.2 Einlesen), mit Hilfe einer einfachen Abfrage ermittelt und in der Darstellung auf eine Zeile reduziert werden können (s. Abb. 5.16).

Weitere Möglichkeiten, um den Java Custom-Viewer den nutzerspezifischen Erfordernissen anzupassen, sind ausführlich in dem ESRI-Handbuch 'Customizing ArcIMS-Java Viewer' beschrieben (ESRI 2001a).

⁹Die vollständige Javascript-Funktion 'showSelect()' ist im Anhang D abgedruckt.


```
var anzahl = 0; //Zahl
var vergleichswert = 0; //Zahl
//Tabellenrumpf
//beginnt ab '1', da Feld[0] die Spaltennamen enthält
//alle Datensätze werden einzeln eingelesen
for (var i=1;i < theList.length; i++) {
  //'theValues[Feld]' enthält je Feld einen Attributwert einer Zeile
  theValues = theList[i].split("::");
  //Vergleich der aktuellen 'CruiseID' mit der des vorhergehenden
  //Datensatzes, wenn ja überspringen
  if (theValues[1] == vergleichswert) {
    continue;
  }
  else {
    vergleichswert = theValues[1];
    //enthält die Anzahl der verschiedenen Datensätze
    anzahl++;
  }
  .
  .
}
```

Abbildung 5.16: Ermittlung der Ergebnisdatensätze gleicher Attributwerte mit einer 'If-Else'-Anweisung'

6 Diskussion der Ergebnisse

Nach dem Aufbau des geographischen Informationssystems und dem Einlesen erster Beispieldatensätze soll nun im Folgenden das Ergebnis vorgestellt werden. Im ersten Teil erfolgt zunächst eine Untersuchung der Funktionalitäten des Java Custom-Viewers hinsichtlich der Analyse des bathymetrischen Datenbestandes sowie der Möglichkeiten zur projektbezogenen Arbeit mit den dargestellten Daten. Dazu werden jeweils exemplarisch einzelne Funktionen vorgestellt. Im zweiten Teil des Kapitels wird auf der Grundlage der in der Zielsetzung (s. Abschnitt 3.1) formulierten Anforderungen eine Ergebniseinschätzung vorgenommen.

6.1 Funktionalität/ Handhabung

In Abbildung 5.13 ist der Java Custom-Viewer mit allen verfügbaren Werkzeugen, die der Analyse des bathymetrischen Datenbestandes dienen, dargestellt. In der Mitte befindet sich die Kartenansicht, in der alle raumbezogenen Daten der aktiven Layer graphisch visualisiert werden. Rechts davon ist im oberen Teil das Übersichtsfenster angeordnet und darunter die Layersteuerung, in der abhängig vom Maßstab die jeweils zur Verfügung stehenden Layer angezeigt werden. Links von der Kartenansicht befindet sich die Werkzeugleiste mit den entsprechenden Analyse-Funktionen.

Innerhalb der Kartenansicht kann mit Hilfe der 'Pan'-Funktion und der 'Zoom'-Funktionen navigiert werden (s. Abb. 6.1). Zur direkten Analyse der visualisierten geographischen Objekte dient das Werkzeug 'Identify', das per Mausklick die vorhandenen Informationen des ausgewählten Objektes anzeigt. Um die Attribute mehrerer Elemente eines aktiven Layers darzustellen, muss die Funktion 'Attributes' aufgerufen werden. Sie ist im vorliegenden Fall das wichtigste Analyse-Werkzeug, da nur über sie der Zugriff auf die Daten des Fileservers möglich ist (s. 5.6). Die Abbildung 6.2 zeigt dazu das Beispiel einer HTML-Tabelle, die nach dem Aufruf der 'Attributes'-Funktion mit den entsprechenden Internet-Links aufgebaut wird.

KAPITEL 6. DISKUSSION DER ERGEBNISSE

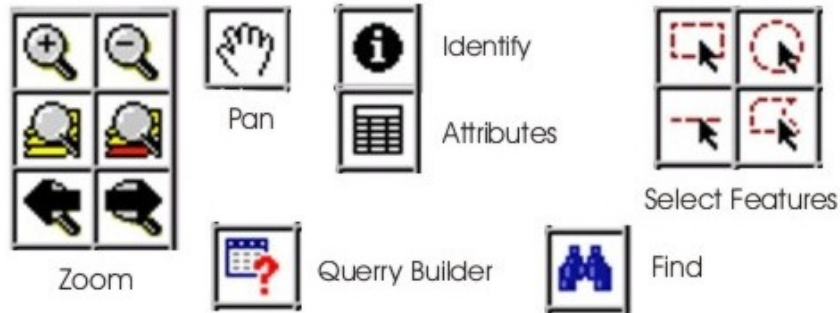


Abbildung 6.1: Die wichtigsten Funktionen zur Analyse des Datenbestandes

CRUISE3

Rec ? of 3	CRUISEID	STARTDATE	ENDDATE	STARHARBOR	ENDHARBOR	INSTITUT	SHIPNAME	CRUISENAME
1	5	24.02.2001	12.04.2001	Punta Arenas	Punta Arenas	AWI	Polarstern	ANT-XVIII/5a
2	9	01.12.2001	21.01.2002	Capetown	Punta Arenas	AWI	Polarstern	ANT-XIX/2
3	12	03.04.2002	05.05.2002	Punta Arenas	Punta Arenas	AWI	Polarstern	ANT-XIX/5

CHIEFSCIENTIST	NAVPLATFORM	DEPTHPLATFORM	METADATA	RAWDATA	REPORT
Gersonde, R.	M2	H2D1P2	antxviii5a.htm	AntXVIII5a	ReportAntXVIII5a.pdf
Jokat, W.	M2	H2D1P2	antxix2.htm	AntXIX2	ReportAntXIX2.pdf
Arntz, W.	M2	H2D1P2	antxix5.htm	AntXIX5	ReportAntXIX5.pdf

Abbildung 6.2: Nach dem Aufruf der 'Attributes'-Funktion wird eine HTML-Tabelle aufgebaut, die über Internet-Links den Zugriff auf die Daten des Fileservers ermöglicht.

Bevor jedoch diese Funktion aufgerufen werden kann, ist es notwendig, die gewünschten Objekte in der Kartenansicht zu selektieren. Dazu dienen entweder die graphischen 'Selektions'-Werkzeuge oder der 'Query Builder' und die 'Find'-Funktion, die eine Auswahl von Objekten anhand der jeweils zugehörigen Eigenschaften erlauben. In Abbildung 6.3 wurde mit Hilfe des 'Query Builders' eine Abfrage generiert, die alle Expeditionen selektieren soll, auf denen eine Hydrosweep-Anlage zum Einsatz kam. Nach der Ausführung des 'Execute'-Befehls sowie dem Aufruf der 'Attributes'-Funktion werden die Ergebnisdatensätze in einer HTML-Tabelle zusammengestellt (s. Abb. 6.2).

Weiterführende Informationen zur Handhabung der aufgeführten Werkzeuge sind in dem ESRI-Handbuch 'Using ArcIMS' zu finden (ESRI (2002e), S. 138 ff).

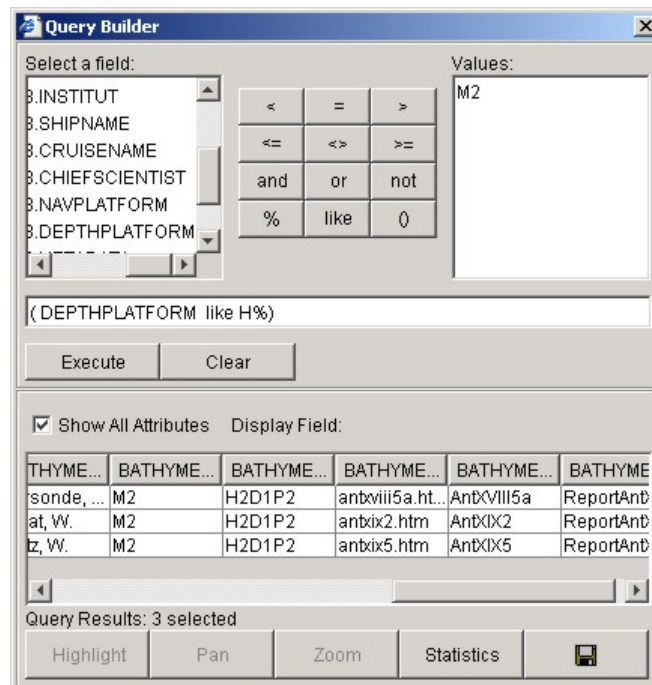


Abbildung 6.3: Abfrage aller Expeditionen, auf denen eine Hydrosweep-Anlage (unabhängig vom Typ -> 'H%') zum Einsatz kam.

Bei der projektbezogenen Arbeit mit den dargestellten Daten kommen vor allem die Vorteile des *Feature Service* gegenüber dem *Image Service* zum Tragen. So hat der Nutzer die Möglichkeit mit Hilfe der 'Layer Properties'-Funktion (s. Abb. 6.4) die Eigenschaften der angezeigten Layer hinsichtlich der Art der Beschriftung sowie der Farbe und Größe der verwendeten Symbole zu verändern. Dies dient vor allem der Übersichtlichkeit, insbesondere wenn zusätzliche Layer, sei es von der lokalen Festplatte, über die Datenbankschnittstelle oder aus dem Internet, mit der 'Add-Layer'-Funktion hinzugeladen werden (s. Abb. 6.5).

KAPITEL 6. DISKUSSION DER ERGEBNISSE

Außerdem können die Eigenschaften aller geladenen Layer in einem *Map Configuration File* gespeichert werden, wodurch es möglich ist, vorgenommene Änderungen am Karteninhalt zu einem späteren Zeitpunkt mit Hilfe der 'Project'-Funktionen erneut aufzurufen. Ebenso kann aber auch der dargestellte Karteninhalt als Bild im JPEG-Format gespeichert werden.

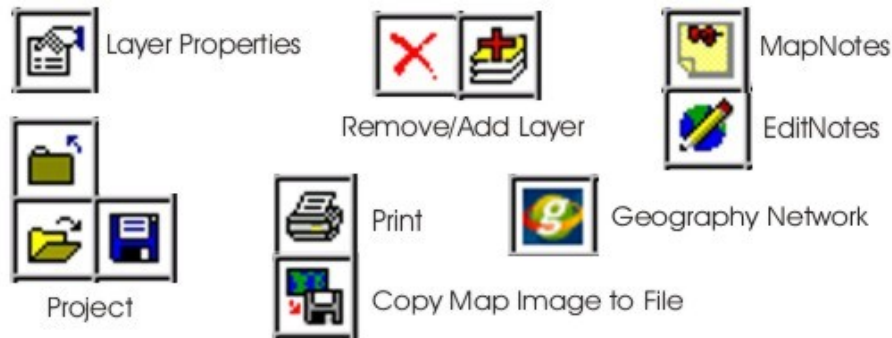


Abbildung 6.4: Wichtige Funktionen zur projektbezogenen Arbeit mit den dargestellten Daten.

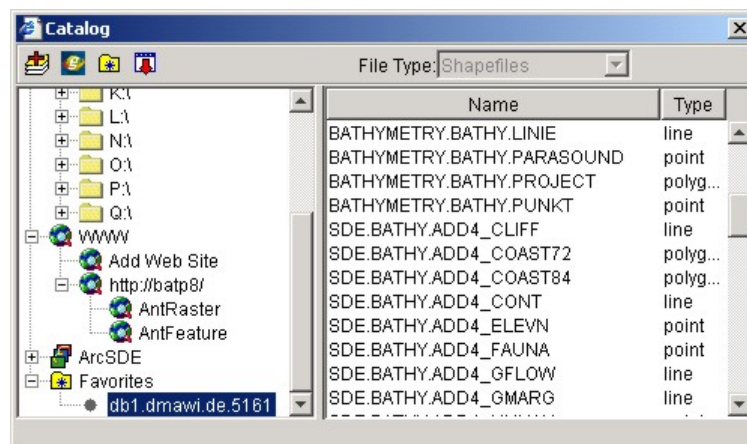


Abbildung 6.5: Hinzufügen von weiteren Daten-Layern (*Shapefiles*, Feature Classes und ausgewählte Bildformate (s. Anhang A.3))

Die beiden bedeutendsten und gleichzeitig umfangreichsten Werkzeuge, die der Java Custom-Viewer bei der Arbeit mit einem *Feature Service* bietet, sind die 'MapNotes'- und die 'EditNotes'-Funktionen. Sie erlauben im Rahmen der Internet/Intranet-Anwendung das Hinzufügen von Text und Graphiken sowie die Veränderung der Attributwerte der einzelnen Daten-Layer. Damit ist die Möglichkeit gegeben über das Internet nicht nur Daten,

sondern auch Ideen auszutauschen (ESRI (2002e), S. 152 ff).

6.2 Ergebniseinschätzung

Die Realisierung der in der Zielsetzung (Abschnitt 3.1) formulierten Anforderungen wird im Folgenden anhand der zugrundeliegenden 3-Ebenen Client/Server Architektur diskutiert, die durch das Datenmodell, den ArcIMS sowie den Java Custom-Viewer beschrieben wird. Das Ziel dabei ist es, die gestellten Anforderungen konkret den genannten Komponenten zuzuordnen, um eine geeignete Einschätzung der Realisierung vornehmen zu können und um absehbare Grenzen aufzuzeigen.

Der Java Custom-Viewer bietet unabhängig vom ausgewählten ArcIMS Service eine Vielzahl von Funktionen, die in der Zielsetzung gefordert waren und die der Analyse des bathymetrischen Datenbestandes dienen. Dazu gehören neben den 'Zoom'-Funktionen und den graphischen 'Selektions'-Werkzeugen auch der 'Query Builder' zur Generierung von inhaltlichen Abfragen sowie die 'Attributes'-Funktion zur Anzeige der Ergebnisdatensätze. Außerdem verfügt der Java Custom-Viewer im Gegensatz zum HTML-Viewer über die beiden Werkzeuge 'EditNotes' und 'MapNotes' sowie über die 'Add Layer'-Funktion. Diese ermöglicht es, zusätzliche Daten in die Kartenansicht zu laden. Allerdings ist eine „on the fly“-Projektion der entsprechenden Daten (im Unterschied zu ArcMap) nicht möglich. Dies setzt voraus, dass die hinzuzufügenden Daten bereits in einer Polar Stereographischen Projektion vorliegen.

Durch den Einsatz des ArcIMS wird der Zugriff auf den bathymetrischen Datenbestand über Internet/Intranet-Anwendungen gewährleistet. Damit wird auch den Nutzern außerhalb der Arbeitsgruppe die Möglichkeit der Datenrecherche gegeben.

Mit der Kombination der beiden ArcIMS Services (*Feature* und *Image*) können Vektor- und Rasterdaten gleichzeitig im Java Custom-Viewer dargestellt werden. Mit Hilfe der 'Layer Properties'-Funktion ist der Nutzer noch nach dem Aufruf der Webseite in der Lage, das Layout der Vektordaten innerhalb der Kartenansicht seinen Erfordernissen anzupassen. Dies ist die Grundlage für die Erstellung übersichtlicher Trackplots mit der 'Copy Map Image to File'-Funktion.

Durch die Verwendung des *Feature Services* und des Java Custom-Viewers wird allerdings die Browserkompatibilität eingeschränkt. Das hat zur Folge, dass die erstellte Webseite nur vom Internet Explorer (Version 4.x oder höher) interpretiert werden kann. Außerdem ist beim ersten Aufruf der Webseite das Herunterladen des Java2-Applets erforderlich (s. Viewer, S. 38).

Die Organisation und Verwaltung der bathymetrischen Daten wird durch Kombination von Fileserver und *Geodatabase* erreicht. Dabei werden alle raumbezogenen Daten in der

Geodatabase gespeichert und entsprechend von ArcSDE verwaltet (s. 4.3.2 ArcSDE). Die beiden zusätzlich angelegten Metadatentabellen 'Cruise' und 'Project', die aufgrund ihres geographischen Bezugs ebenfalls im Java Custom-Viewer visualisiert werden können, ermöglichen es, den Datenbestand hinsichtlich zeitlicher, inhaltlicher und geographischer Aspekte zu untersuchen. Außerdem wird über diese Tabellen mit Hilfe der 'Attributes'-Funktion die Verknüpfung zu den Roh-, Projekt- und Metadaten, die auf dem Fileserver verwaltet werden, realisiert. Damit ist der Zugriff auf den gesamten bathymetrischen Datenbestand gewährleistet.

Allerdings ist das Herunterladen von raumbezogenen Daten im Rahmen der Internet/Intranet-Anwendung ein sehr zeitintensiver Prozess und somit nur begrenzt möglich. Alle Abfragen, deren Ergebnismenge mehr als 1000 Datenzeilen¹ umfassen könnten (z.B. die Abfrage der Tiefendaten für ein bestimmtes Gebiet), sollten daher statt im Java Custom-Viewer in ArcMap durchgeführt werden. Dies bietet den Vorteil, dass die Daten in einem geeigneten Format (z.B. *Shapefile*) gespeichert werden können und damit zur direkten Weiterverarbeitung in ArcMap zur Verfügung stehen.

Wie in Abbildung 5.1 dargestellt, wurde das zugrundeliegende Datenmodell so konzipiert, dass Daten anderer Fachbereiche, sofern sie über einen Raumbezug verfügen, ebenso verwaltet werden können, wie die Daten der beiden anderen Forschungsschiffe des AWIs.

Mit der Kombination aus Fileserver und *Geodatabase* sowie der Anpassung der Funktionalitäten des Java Custom-Viewers konnten die gestellten Anforderungen in ein System umgesetzt werden, das die Verwaltung des bathymetrischen Datenbestandes erlaubt und den Zugriff über eine benutzerfreundliche webbasierte graphische Oberfläche ermöglicht.

¹Anhand der Erfahrungen des Autors geschätzte Größenordnung.

7 Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit wurde ein geographisches Informationssystem geschaffen, das einerseits die Möglichkeit bietet den bathymetrischen Datenbestand zu verwalten und andererseits diesen für viele Nutzer über eine Internet/Intranet-Anwendung zugänglich macht.

Dazu war zunächst eine Einarbeitung in die bathymetrische Datengewinnung- und aufbereitung erforderlich, um die vorhandenen Datenformate hinsichtlich ihres Inhaltes und damit hinsichtlich ihrer Funktion innerhalb des Informationssystems strukturieren zu können. Außerdem war es notwendig das derzeitige Verwaltungssystem sowie die sich daraus ergebenden Probleme des Dateizugriffs und der Datenaktualisierung zu analysieren. Auf dieser Grundlage wurden die Anforderungen an ein neues System zur Verwaltung des bathymetrischen Datenbestandes formuliert.

Bevor jedoch ein geeignetes Datenmodell konzipiert werden konnte, war eine umfangreiche und intensive Auseinandersetzung mit den vorhandenen ArcGIS-Produkten der Firma ESRI erforderlich. Dabei standen besonders die Funktionalitäten der ArcSDE und des Internet Map Servers (ArcIMS) im Vordergrund.

Nach dem Aufbau der Datenbank und dem Anlegen des Fileservers konnten erste Beispieldatensätze mit Hilfe der ArcGIS Desktop-Funktionen und den beiden zusätzlich erstellten C++-Programmen eingelesen werden. Im Anschluß erfolgte die Erstellung von standardkonformen Metadaten sowie die Anlage der Webseite mit entsprechendem ArcIMS Viewer. Als letztes wurde die Verknüpfung zwischen *Geodatabase* und Fileserver mit Hilfe der Javascript-Funktion 'ShowSelect' realisiert.

Das Ergebnis ist ein funktionsfähiges geographisches Informationssystem, das die Verwaltung des bathymetrischen Datenbestandes ermöglicht. Allerdings gelingt dies nur, wenn das System auch mit „Leben“ gefüllt wird. Dazu ist es notwendig den vorhandenen Datenbestand entsprechend aufzuarbeiten und in das System zu importieren. Für zukünftige Expeditionen sollten Verfahrensabläufe aufgestellt werden, die eine einheitliche und umfassende Dokumentation der jeweiligen Messanordnung erlauben, um die Nacharbeit, die für die Integration der aufgezeichneten Daten in das neue Informationssystem erforderlich ist, möglichst gering zu halten.

KAPITEL 7. ZUSAMMENFASSUNG UND AUSBLICK

Aufbauend auf die vorliegende Arbeit ist die Entwicklung von Funktionen denkbar, die eine „on the fly“-Generalisierung für die in der Kartenansicht des Java Custom-Viewers darzustellenden Daten ermöglichen, um eine doppelte Datenhaltung, wie sie bei den 'Cruise'-Tabellen in dieser Arbeit auftritt, zu vermeiden. Außerdem wäre zu untersuchen, ob innerhalb der Internet/ Intranet-Anwendung ein Umprojizieren der visualisierten Daten zwischen verschiedenen Koordinatensystemen realisierbar ist, um z.B. das Hinzuladen weiterer Daten zu vereinfachen. Damit würde die praktische Umsetzung der Idee, von der webbasierten Erzeugung bathymetrischer Karten, in nahe Zukunft rücken.

Literatur

- Bill, R. und D. Fritsch (1991). *Grundlagen der Geo-Informatinssysteme, Band 1*. Wichmann Verlag, Karlsruhe.
- Christen, S. (1999). *Aufbereitung, Visualisierung und Interpretation einer bathymetrischen Vermessung im Süd Sandwich Tiefseeegraben*. Diplomarbeit, Fachhochschule Hamburg.
- Dickmann, F. (2001). *Web-mapping und Web-gis*. Braunschweig: Westermann Druck GmbH.
- Diepenbroek, M., H. Grobe, M. Reinke, R. Schlitzer und R. Sieger (1999). *PANGAEA - an Information System for Environmental Sciences*. Bremerhaven, Deutschland: Alfred-Wegener-Institut. <http://www.pangaea.de/Paper/CG/>; Datum des Zugriffs: 23.02.2004.
- ESRI (2001a). *Customizing ArcIMS-Java Viewer* (2 ed.). Redlands, USA. Manual; Printed in the United States of America.
- ESRI (2001b). *Using ArcCatalog*. Redlands, USA. Manual; Printed in the United States of America.
- ESRI (2002a). *ArcSDE Configuration and Tuning Guide for Microsoft SQL Server*. Redlands, USA. Manual; Printed in the United States of America.
- ESRI (2002b). *Building a Geodatabase*. Redlands,USA. Manual; Printed in the United States of America.
- ESRI (2002c). *Metadata and GIS*. Redlands, USA. http://downloads.esri.com/support/whitepapers/ao_/metadata-and-gis.pdf; Datum des Zugriffs: 16.02.2004.
- ESRI (2002d). *Understanding ArcSDE*. Redlands,USA. Manual; Printed in the United States of America.
- ESRI (2002e). *Using ArcIMS*. Redlands, USA. Manual; Printed in the United States of America.

LITERATUR

- ESRI (2003a). *ArcIMS 4 Architecture and Functionality*. Redlands,USA.
http://downloads.esri.com/support/whitepapers/ims_/arcims4_architecture.pdf;
Datum des Zugriffs: 06.03.2004.
- ESRI (2003b). *Was ist ArcGIS?* ESRI Geoinformatik GmbH, Deutschland.
http://www.esri-germany.de/downloads/papers/was_ist_arcgis_8_dt_print.pdf;
Datum des Zugriffs: 02.03.2004.
- FGDC (1998). *Content Standard for Digital Geospatial Metadata*. Virginia,USA: FGDC.
- Frahm, A. (2003). Qualitative Untersuchungen einer ausgeprägten Tiefseerinnenstruktur am ostgrönländischen Kontinentalhang. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg.
- Garcia, M. F., J. Reding, E. Whalen und S. A. DeLuca (2000). *Microsoft SQL Server 2000 - Das Handbuch*. Unterschleißheim, Deutschland: Microsoft Press Deutschland.
- Gerchow, R. (2001). Anschlußplan für die Geräte von PODAS. Systembeschreibung, FIELAX GmbH, Bremerhaven, Deutschland.
- Greitemann-Hackl, A. (2001). (Schiffs)Positionssysteme und Positionssensoren auf dem FS „Polarstern“. Systembeschreibung, Alfred-Wegener-Institut für Polar- und Meeresforschung, Bremerhaven, Deutschland.
- Grobe, H., M. Diepenbroek und M. Reinke (1997). *PANGAEA - ein Informationssystem für die Meeresforschung*. Bremerhaven, Deutschland: Alfred-Wegener-Institut.
<http://www.pangaea.de/Paper/DGM/DGM.pdf>; Datum des Zugriffs: 23.02.2004.
- Iffland, A. (2004). Aufarbeitung und Visualisierung einer bathymetrischen Vermessung in Verbindung mit Seismogrammen der Sedimentechographie. Diplomarbeit, Fachhochschule Hamburg.
- Krocker, R. (2002). *Bathymetrie-Datenbank*. Bremerhaven, Deutschland: AWI.
- Kurtze, C. (2003). Aufbau eines Auskunftssystems für das Österreichische Archäologische Institut auf der Basis des ArcIMS. Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden.
- Rottmann, E., H. W. Schenke und T. Schöne (1994). Erfahrungen mit dem DGPS-system SKYFIX auf „FS Polarstern“. *Dokumentation zum 9. Hydrographentag in Hamburg*.
- Schenke, H. W. (1990). Die Fächersonaranlage Hydrosweep (Ein neues System zur flächenhaften Vermessung des Meeresbodens). *Die Geowissenschaften* 6, 161–169.
- Schenke, H. W. (1992). GPS und Fächersonarmessungen mit Hydrosweep auf FS Polarstern. *Zeitschrift für Vermessungswesen* 8/9, 599–615.

- Spieß, V. (1993). *Digitale Sedimentechographie - Neue Wege zu einer hochauflösenden Akustostratigraphie*. Universität Bremen, Fachbereich Geowissenschaften, Bremen: unveröffentlichte Dissertation.
- Transtec (2003). *Produktkatalog*. Tübingen, Deutschland: Transtec AG.
- Werum (2001). *PODAS*. Lüneburg, Deutschland: Werum Software & System AG.

Danksagung

Nachdem nun die druckreife Arbeit vor mir liegt, möchte ich all denjenigen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Für die Schaffung der äußeren Rahmenbedingungen zur praktischen Durchführung der Diplomarbeit gilt mein ausdrücklicher Dank

Herrn Dr. H. W. Schenke und Herrn Dr. M. Klenke

vom Alfred-Wegener-Institut, sowie meinen beiden Gutachtern von der Fachhochschule in Dresden

Herrn Prof.Dr.-Ing. F. Schwarzbach und Herrn Prof.Dr.-Ing. G. Reppchen

Weiterhin danke ich Fred, für die vielen Antworten zu den Bits und Bytes der Bathymetrie, sowie Andreas, Rike, Ralph und Tanja für zahlreiche Anregungen und Gespräche jeglicher Art.

Außerdem möchte ich meiner Freundin Britta und meinem Bruder Christoph für die moralische Unterstützung und für das viele Korrekturlesen danken.

Natürlich gilt mein Dank auch all den Beteiligten, die ich nicht explizit aufgeführt habe.

A Anhang A

Die *Geodatabase*

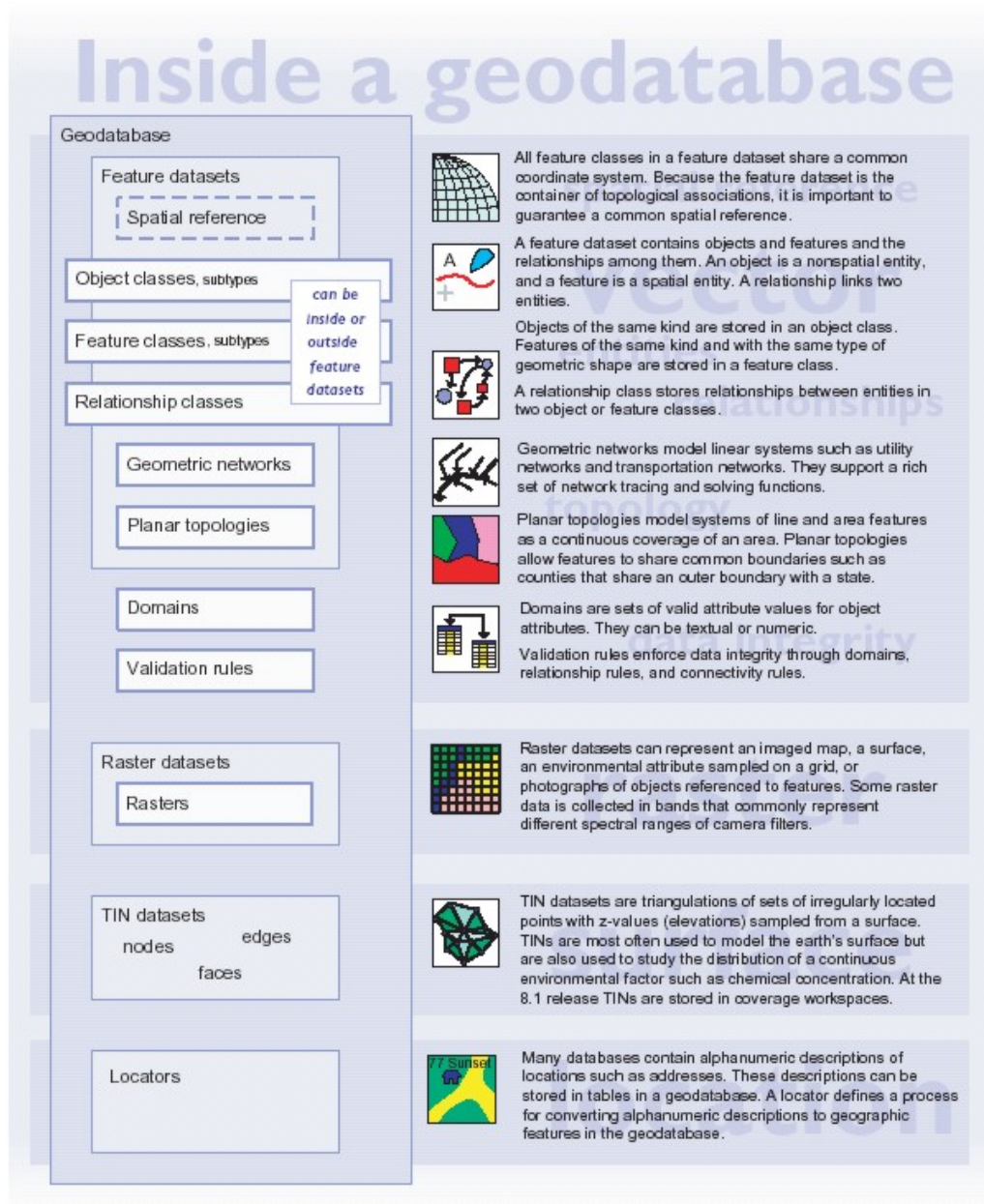


Abbildung A.1: Die Objekte einer *Geodatabase* (ESRI (2002d), S. 13)

Personal/ArcSDE Geodatabase

	DBMS	Client/Server	Objects	Long Transactions*	Editors	C or Java API	Raster	Size
Multiuser Geodatabases	Oracle, Microsoft SQL Server, Informix, IBM DB2	Yes	Yes	Yes	1 or more	Yes	Yes	Unlimited
Personal Geodatabase	Microsoft Jet	No	Yes	No	1 only	No	No	Up to 2 Gb

*Database transactions spanning multiple edit sessions

Abbildung A.2: Vergleich von *Personal Geodatabase* und *ArcSDE (Multiuser) Geodatabase* (ESRI 2002d, S. 12)

Von ArcIMS unterstützte Bildformate

Short Name	Descriptive Name	Image File Extensions	World File Extensions
ADRG	ARC Digitized Raster Graphics	*.img, *.ovr, *.l	
<u>ASRP</u> ¹	DIGEST ARC Standardized Raster Product	*.img, *.ovr, *.l	
BIL	Band Interleaved by Line	*.bil	*.blw
BIP	Band Interleaved by Pixel	*.bip	*.bpw
BMP	Windows Bitmap	*.bmp	*.bpw
BSQ	Band Sequential	*.bsq	*.bqw
CADRG	Compressed ARC Digitized Raster Graphics	*.* (no standard)	*.* (no standard)
CIB	Controlled Image Base	*.* (no standard)	
ERDAS	ERDAS [®] GIS or LAN	*.gis, *.lan	*.gsw, *.lnw
GeoTIFF	TIFF with Geo Header	*.tif	
GIF	Graphics Interchange Format	*.gif	*.gfw
<u>GRID</u> ²	ArcGrid™	*.*(no standard)	
<u>IMAGINE</u> ¹	ERDAS IMAGINE™	*.img	
IMPELL	Impell Bitmap	*.rlc	*.rlw
JFIF, JPG	JPEG	*.jpg	*.jgw
<u>MrSID</u> ²	Multiresolution Seamless Image Database	*.sid	*.sdw
NITF	National Imagery Transfer Format	*.ntf	*.nfw
PNG	Portable Network Graphics	*.png	*.pgw
SUN	SunRaster™ File	*.sun, *.ras	*.snw, *.rsw
<u>TIFF</u> ³	Tag Image File Format	*.tif, *.tff	*.tfw
<u>USRP</u> ¹	DIGEST UTM/UPS Standardized Raster Product	*.img, *.ovr	

¹ Format is not supported on UNIX or Linux[®].

² Format is supported on Windows and Sun[®] Solaris but not on HP-UX, IBM[®] AIX[®], Linux, or SGI IRIX.

³ CCITT4 compressed TIFFs are not supported.

Abbildung A.3: Von ArcIMS unterstützte Bildformate mit den dazu erforderlichen „world file“-Erweiterungen (aus der Hilfe des ArcIMS Author)

Funktionsumfang der ArcIMS Services

Data Types	Data Format	Image	Feature	ArcMap
Coverages	ArcInfo Coverages	No	No	Yes
	PC ARC/INFO® Coverages	No	No	Yes
	ArcSDE for Coverages	Yes	Yes	Yes
ArcSDE •SQL Server™ •Informix® •DB2® •Oracle®	ArcSDE Features	Yes	Yes	Yes
	ArcSDE—Versioned Layers	No	No	Yes
	ArcSDE Multiraster and 32-Bit Raster (Oracle)	Yes	No	Yes
	ArcSDE Raster (SQL Server, Informix, DB2)	Yes	No	Yes
CAD	DWG	No	No	Yes
	DXF	No	No	Yes
	DGN	No	No	Yes
Raster	Image Catalog (Raster Catalog)	Yes	No	Yes
	ADRG Image (.IMG)	Yes	No	Yes
	ADRG Overview (.OVR)	Yes	No	Yes
	ADRG Legend (.LGG)	Yes	No	Yes
	Band Interleaved by Line (.BIL)	Yes	No	Yes
	Band Interleaved by Pixel (.BIP)	Yes	No	Yes
	Band Sequential (.BSQ)	Yes	No	Yes
	Bitmap—Windows (.BMP)	Yes	No	Yes
	Controlled Image Base (.CIB)	Yes	No	Yes
	CADRG (.CRG)	Yes	No	Yes
	DIGEST ARC Standardized Raster Product (ASRP)	Yes	No	No
	DIGEST UTM/UPS Standardized Raster Product (USRP)	Yes	No	No
	DTED Level 1 and 2 (.DT1)	No	No	Yes
	ERDAS® Image (.IMG)	Yes	No	Yes
	ERDAS 7.5 Lan (.LAN)	Yes	No	Yes
	ERDAS 7.5 GIS (.GIS)	Yes	No	Yes
	ERDAS Raw (.RAW)	No	No	Yes
	ER Mapper (.ERS)	No	No	Yes
	ESRI GRID	Yes	No	Yes
	ESRI GRID Stack	No	No	Yes
	Graphic Interchange Format, GIF (.GIF)	Yes	No	Yes
Impell Bitmap (IMPELL)	Yes	No	Yes	
Raster (cont.)	JPEG (.JPG)	Yes	No	Yes
	MrSID®—LizardTech (.sid)	Yes	No	Yes
	National Image Transfer Format (.NTF)	Yes	No	Yes
	Portable Network Graphics (.PNG)	Yes	No	Yes
	SunRaster File (.SUN)	Yes	No	No
	Tagged Image File Format (.TIF)	Yes	No	Yes
	TIFF with Geo Header (.TIF)	Yes	No	Yes
Other	Annotation Layers	No	No	Yes
	TIN	No	No	Yes
	VPF	No	No	Yes

Abbildung A.4: Unterstützte Datentypen der ArcIMS Services (ESRI (2003a), S. 12/13)

B Quellcode

B.1 Programm 'NakKonverter'

(Seite 1/6)

```

//Programmbeschreibung:
//Eingabe: Programmaufruf, Listing-Datei, Fahrabschnittsname (*), CruiseID (~)
//Ausgabe: *DWS.txt,*Para.txt,*gen.
//Programm befindet sich im gleichen Ordner wie Listing-Datei und Nak-Dateien
//Listing-Datei: auf jeder Zeile steht eine Nak-Datei;
//CruiseID: Datenbankabfrage durchführen, welches die letzte CruiseID ist und dann hier
//nächst mögliche vergeben
//Programmablauf: jede im Listing angegebene Datei wird zeilenweise ausgelesen;
//1. Überprüfung ob die aktuelle Zeile im angegebenen Koordinatenbereich liegt (siehe
//'breite') und Vergleich der aktuellen Koordinate mit der vorhergehenden; wenn diese
//der vorhergehenden entspricht, wird die aktuelle nicht mit übernommen
//2. Auslesen der DWS-Tiefen mit Position und Zeitstempel - Tiefen erhalten ein Minus
//(im Gegensatz zu Nak-Format)
//3. Auslesen der Parasound-Tiefen mit Position und Zeitstempel - Tiefen erhalten ein
//Minus (im Gegensatz zu Nak-Format)
//4. Erstellen der GEN-Datei für die Erzeugung von Linie-Coverages mit ArcToolbox, es
//werden 3 Linien mit unterschiedlicher Auflösung angelegt; 100~ - 10 min; 200~ - 2 h;
//300~ - 8 h (d.h. z.B. aller 10 min ein Koordinatenwert)
//
//Plausibilitätstest: PODAS speichert solange die gleiche Parasound-Tiefe (mitunter gilt
//dies auch für die DWS-Tiefe) bis ein neuer Wert vom entsprechenden Messsystem gesendet
//wird. Dies entspricht aber keinesfalls dem jeweilig möglichen Messimpuls: Im Programm
//werden daher zu Beginn zwei Parameter definiert (wdhDws bzw. wdhPara), die angeben,
//wie oft sich ein Tiefenwert bei fortlaufender Positionsveränderung wiederholen darf,
//bevor er als tatsächlich gemessen mit in die Ausgabedatei übernommen wird. Ziel ist es
//dabei, die 'aufgefüllten' Tiefenwerte zu eliminieren, um die Ausgabedatei möglichst
//klein zu halten.
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include <fstream.h>
#include <iostream>
#include <vector>
#include <string.h>
#include <stdio.h>
#include <math.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

struct PARASOUND{
    CString datetime;
    double lat;
    double lon;
    double depth;
    int cruiseID;
};

struct DWS{
    CString datetime;
    double lat;
    double lon;
    double depth;
    int cruiseID;
};

```

Programm 'NakKonverter' (Seite 2/6)

```

struct TRACK{
    double lat;
    double lon;
};

struct INHALT{
    CString datetime;
    double lat;
    double lon;
    double dwsdepth;
    double paradepth;
};

int main(int argc, char* argv[])
{
    if (argc < 4) {
        printf("Zu wenig parameter: Programm, Listing, Fahrtname, cruiseID \n");
        return 1;
    }

    char zeile[130]; // '130' angenommener Wert für max. Zeilenlänge
    char namenzeile[20]; // '15' Länge eines Dateinamen in der Listing-Datei
    char *token;

    //Parameter
    double breite = -50.0; //nur Koordinaten werden eingelesen, die südlich dieser Breite
                          //liegen

    //Problem: PODAS gibt für jede Sekunde eine Tiefe aus --> es erscheinen mehrmals
    //die gleichen Tiefen
    //Wie oft darf die gleiche Tiefe erscheinen, bevor sie erneut in die Ausgabe übernommen
    //wird
    int wdhdws = 4; //Wiederholungswert für DWS
    int wdhpars = 11; //Wiederholungswert für Parasound

    unsigned long c,j,anz;

    CString feld[13];

    CString cruiseString, fahrtname, dateiname, zeileninhalt;
    CString DateiDws, DateiPara, DateiTrack; //Ausgabedateinamen
    CString datum, punktzeile;

    int num = 0;
    int cruiseID;

    double templat, templon;

    //Variablen der einzelnen Strukturen
    PARASOUND Para;
    DWS Dws;
    INHALT Inhalt;
    TRACK Track;

    //Folgende Vektoren enthalten alle gültigen Werte eines Fahrtabschnittes
    //(Dateiübergreifend)
    std::vector<TRACK> Track10list;
    std::vector<TRACK> Track120list;
    std::vector<TRACK> Track480list;

    //Liest Namen der Listendatei ein
    ifstream liste(argv[1]);

```


Programm 'NakKonverter' (Seite 3/6)

```

//Fahrtnamen einlesen und Erstellen der Ausgabedateinamen
fahrtname = argv[2];

DateiDws = fahrtname + "Dws.txt";
DateiPara = fahrtname + "Para.txt";
DateiTrack = fahrtname + ".gen";

//CruiseID wird eingelesen
cruiseID = atoi(argv[3]);
cruiseString = argv[3]; //Für die Ausgabe der *.gen Datei

//Anlegen der Ausgabedateien
ofstream txtDateiDws(DateiDws);
ofstream txtDateiPara(DateiPara);

anz =0; //Anzahl der Dateien

//////////Durchlauf des Datei-Listings ////////////////////////////////////////////
while (!liste.eof()) {
    anz++;
    liste.getline(namenzeile,20); //liest Dateinamen ein
    dateiname=namenzeile;
    if (dateiname.GetLength()<4) continue; //Falls Dateiname zu kurz

    ifstream datei(namenzeile);

    //Für jede Datei wird ein neuer Vektor angelegt
    std::vector<INHALT> Inhaltlist;

    ////////////Auslesen einer einzelnen Datei in einen Vektor//////////////////////////////////////////
    c=0; //Anzahl der ausgelesenen Zeilen insgesamt (auch die, die nicht
        //südlich 50° Süd liegen)
    templat=0.0; templon=0.0;
    while (!datei.eof()) {

        datei.getline(zeile,130);
        zeileninhalt=zeile;
        if (zeileninhalt.GetLength()<4) continue; //Falls Zeile zu kurz

        c++; //Zeilenzähler

        token=strtok(zeile, " ,\t"); //überschreibt gefundene delimiter und token
        //mit '\0'
        //bei jedem weiteren Aufruf von 'strtok' wird der verbleibende Rest
        //aufgerufen; siehe auch Hilfe

        j=0;
        while (token != NULL) {

            feld[j]=token;
            j++; //zählt die Tokens je Zeile
            // Get next token:
            token = strtok( NULL, " ,\t" );
        }

        //Nur Werte südlich von 50°
        if (atof(feld[3])<breite) {
            if ((atof(feld[3])!=templat) || (atof(feld[4])!=templon)) {

                Inhalt.datetime = feld[0] + " " +feld[1];
                Inhalt.lat = atof(feld[3]);
                Inhalt.lon = atof(feld[4]);
            }
        }
    }
}

```


Programm 'NakKonverter' (Seite 4/6)

```

//aus "positiven" Tiefen werden wirkliche Tiefen gemacht (* (-1))
Inhalt.dwsdepth = ((atof(feld[7]))*-1);
Inhalt.paradept = ((atof(feld[8]))*-1);

//Struktur Inhalt in den Vektor schreiben
Inhaltlist.push_back(Inhalt);
templat = atof(feld[3]);
templon = atof(feld[4]);
}
}

} //Einlesen der ersten Datei im Listing beendet

//////////////////////////////////// Ausgabe der einzelnen Dateien //////////////////////////////////////
double tempTiefe =0; //Vergleichswert
int wd =0; //Wie oft darf eine Tiefe auftreten,
//ohne dass sie jedesmal gemessen wurde

//Ausgabe der DWS-Tiefen
for (int d=0;d<Inhaltlist.size();d++) {
    INHALT Zeile = Inhaltlist[d];
    if (Zeile.dwsdepth<0) {
        if ((Zeile.dwsdepth == tempTiefe) && (wd<wdhDws)) {
            wd++;
        } else {
            //die Tiefe hat sich innerhalb des Wiederholungszeitraumes (wdh) verände:
            Dws.datetime = Zeile.datetime;
            Dws.lat = Zeile.lat;
            Dws.lon = Zeile.lon;
            Dws.depth = Zeile.dwsdepth;
            Dws.cruiseID = cruiseID;

            //Koordinaten werden rausgeschrieben und zu einem String zusammengesetzt
            punktzeile.Format(" ;%s;%9.5f;%9.5f;%8.2f;%d",
                Dws.datetime,Dws.lat,Dws.lon,Dws.depth,Dws.cruiseID);
            txtDateiDws.write(punktzeile,punktzeile.GetLength());
            endl(txtDateiDws);
            //
            tempTiefe = Zeile.dwsdepth;
            wd=0;
        }
    }
}

tempTiefe=0;wd=0;
//////////////////////////////////// Ausgabe der Parasound-Tiefen //////////////////////////////////////
for (int p=0;p<Inhaltlist.size();p++) {
    INHALT Zeile = Inhaltlist[p];
    if (Zeile.paradept<0) {
        if ((Zeile.paradept == tempTiefe) && (wd<wdhPara)) {
            wd++;
        } else {
            //die Tiefe hat sich innerhalb des Wiederholungszeitraumes
            //(wdh) verändert, oder 'wdh' ist überschritten
            Para.datetime = Zeile.datetime;
            Para.lat = Zeile.lat;
            Para.lon = Zeile.lon;
            Para.depth = Zeile.paradept;
            Para.cruiseID = cruiseID;
        }
    }
}

```

Programm 'NakKonverter' (Seite 5/6)

```

        //Koordinaten werden rausgeschrieben und zu einem String zusammengesetzt
        punktzeile.Format(" %s;%9.5f;%9.5f;%8.2f;%d",
            Para.datetime,Para.lat,Para.lon,Para.depth,Para.cruiseID);
        txtDateiPara.write(punktzeile,punktzeile.GetLength());
        endl(txtDateiPara);
        //
        tempTiefe = Zeile.paradept;
        wdh=0;
    }
}

////////////////////////////////////
//Alle Daten, die zur Erzeugung der gen.Datei notwendig sind werden in einen
//Vektor geschrieben
for (int i=0;i<Inhaltlist.size();i++) {
    // aller 10 minuten
    if (i%600 == 0) {
        INHALT Zeile = Inhaltlist[i];
        Track.lat = Zeile.lat;
        Track.lon = Zeile.lon;
        Track10list.push_back(Track);
        //aller 2 stunden
        if (i%7200 == 0) { //
            Track120list.push_back(Track);
            //aller 8 stunden
            if (i%28800 == 0) {
                Track480list.push_back(Track);
            }
        }
    }
}

    printf("%s \n",dateiname);
} //Einlesen aller Dateien abgeschlossen

txtDateiDws.close();
txtDateiPara.close();
//ausgabe.close();
printf("Dateien werden erstellt...\n");

//Erstellen der Tracklinien-Datei //////////////////////////////////////
ofstream genDatei(DateiTrack);
CString index;

index = "100" + cruiseString;
genDatei.write(index,index.GetLength());
endl(genDatei);

//Latitude und Longitude müssen noch vertauscht werden
for (int t1=0;t1<Track10list.size();t1++) {
    TRACK TrackZeile = Track10list[t1];
    punktzeile.Format(" %9.5f %9.5f",TrackZeile.lon,TrackZeile.lat);
    genDatei.write(punktzeile,punktzeile.GetLength());
    endl(genDatei);
}
genDatei.write("END",3);endl(genDatei);
index = "200" + cruiseString;
genDatei.write(index,index.GetLength());
endl(genDatei);

```

Programm 'NakKonverter' (Seite 6/6)

```
for (int t2=0;t2<Track120list.size();t2++) {
    TRACK TrackZeile = Track120list[t2];
    punktzeile.Format("    %9.5f    %9.5f",TrackZeile.lon,TrackZeile.lat);
    genDatei.write(punktzeile,punktzeile.GetLength());
    endl(genDatei);
}
genDatei.write("END",3);endl(genDatei);
index = "300" + cruiseString;
genDatei.write(index,index.GetLength());
endl(genDatei);
for (int t3=0;t3<Track480list.size();t3++) {
    TRACK TrackZeile = Track480list[t3];
    punktzeile.Format("    %9.5f    %9.5f",TrackZeile.lon,TrackZeile.lat);
    genDatei.write(punktzeile,punktzeile.GetLength());
    endl(genDatei);
}
genDatei.write("END",3);endl(genDatei);
genDatei.write("END",3);endl(genDatei);

////////////////////////////////////

genDatei.close();
printf("Erfolgreich beendet!");

return 1;
}
```

B.2 Programm 'AsciiKonverter'

(Seite 1/2)

```
//Programmbeschreibung:
//Eingabe: Programmaufruf, Listing-Datei, Fahrtabschnittsname (*), CruiseID (~)
//Ausgabe: *.txt
//Programm befindet sich im gleichen Ordner wie Listing-Datei und ASCII-Dateien
//Listing-Datei: auf jeder Zeile steht eine Nak-Datei;
//CruiseID: Datenbankabfrage durchführen, welches die letzte CruiseID ist und dann hier
//nächst mögliche vergeben
//Programmablauf: jede im Listing angegebene Datei wird zeilenweise ausgelesen;
//1. Überprüfung ob die aktuelle Zeile im angegebenen Koordinatenbereich liegt (siehe
//'breite'); nur gültige Werte werden übernommen
//2. Alle eingelesenen Daten werden in einer einzigen (!) Datei ausgegeben, um das
//Einlesen in die Access-Datenbank zu erleichtern;

#include "stdafx.h"
#include <iostream>
#include <fstream.h>
#include <vector>
#include <string.h>
#include <stdio.h>
#include <math.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

void main(int argc, char* argv[])
{
    if (argc < 3) {
        printf("Zu wenig parameter: Programm, Listing, cruiseID \n");
        return;
    }

    double breite = -50.0;
    char zeile[100]; // '130' angenommener Wert für max. Zeilenlänge
    char namenzeile[20]; // '20' Länge eines Dateinamen in der Listing-Datei
    char *token;

    CString feld[6];
    unsigned long c,j,anz;
    CString cruiseString, fahrtname, dateiname, datenzeile, punktzeile;
    CString zeileninhalt, AusDatei;
    int cruiseID;

    //Liest Namen der Listendatei ein
    ifstream liste(argv[1]);

    //Fahrtnamen einlesen und Erstellen der Ausgabedateinamen
    fahrtname = argv[2];

    AusDatei = fahrtname + "Dws.txt";
```


Programm 'AsciiKonverter' (Seite 2/2)

```

//CruiseID wird eingelesen
cruiseID = atoi(argv[3]);

//Anlegen der Dateien
ofstream ausgabe(AusDatei);

anz =0; //Anzahl der Dateien

////////// Durchlauf der Listing-Datei ////////////////////////////////////////////

while (!liste.eof()) {

    liste.getline(namenzeile,20);//liest Dateinamen ein
    dateiname=namenzeile;
    if (dateiname.GetLength()<4) continue; //Falls Dateiname zu kurz
    anz++;
    ifstream datei(namenzeile);

    c=0; //Anzahl der ausgelesenen Zeilen insgesamt
        //(auch die, die nicht südlich 50° Süd liegen)
    while (!datei.eof()) {

        datei.getline(zeile,130);
        zeileninhalt=zeile;
        if (zeileninhalt.GetLength()<4) continue; //Falls Zeile zu kurz

        c++;//Zeilenzähler
        token=strtok(zeile," \\t");//überschreibt gefundene delimiter und token mit
            //'\\0'; bei jedem weiteren Aufruf von 'strtok' wird der verbleibende Rest
            //'aufgerufen; siehe auch Hilfe
        j=0;
        while (token != NULL) {

            feld[j]=token;
            j++;//zählt die Tokens je Zeile
            // Get next token:
            token = strtok( NULL, " \\t" );
        }

        //Nur Werte südlich von 50°Süd
        if (atof(feld[3])<breite) {

            CString datetime = feld[0] + " " +feld[1];
            int beam = atoi(feld[2]);
            double lat = atof(feld[3]);
            double lon = atof(feld[4]);
            double depth = atof(feld[5]);

            //Koordinaten werden rausgeschrieben und zu einem String zusammengesetzt
            //und entsprechend formatiert
            punktzeile.Format(" ;%s;%d;%10.6f;%10.6f;%9.3f;%d",
                datetime,beam,lat,lon,depth,cruiseID);
            ausgabe.write(punktzeile,punktzeile.GetLength());
            endl(ausgabe);
        }

    }//Einlesen der ersten Datei im Listing beendet
    printf("%s \\n",dateiname);
}
ausgabe.close();
printf("%d Dateien erfolgreich verkettet \\n",anz);
}

```


C Metadaten

Metadaten von ANT XVIII/5a (Seite 1/5)

Expedition Ant XVIII/5a

Metadata also available as

Metadata:

- [Identification Information](#)
 - [Data Quality Information](#)
 - [Spatial Data Organization Information](#)
 - [Spatial Reference Information](#)
 - [Entity and Attribute Information](#)
 - [Distribution Information](#)
 - [Metadata Reference Information](#)
-

Identification Information:

Citation:

Citation Information:

Originator:

Bathymetrie, Alfred-Wegener-Institut für Polar- und Meeresforschung

Publication Date: 2004

Title: Expedition Ant XVIII/5a

Geospatial Data Presentation Form: vector digital data

Publication Information:

Publication Place: Bremerhaven

Publisher: Alfred Wegener Institut für Polar- und Meeresforschung

Online Linkage:

Server=db1.dnawi.de; Service=5161; Database=Bathymetry; User=bathy;

Version=sde.DEFAULT

Description:

Abstract:

Die zugrundeliegenden SURF-Daten wurden auf der Expedition Ant XVIII/5a mit dem Forschungsschiff "FS Polarstern" in der Antarktis aufgezeichnet. Dabei kam das Fächersonarsystem das Hydrosweep DS2 der ATLAS Hydrographics GmbH zum Einsatz.

Das Gebiet erstreckt sich von 127°W - 48°W und von 51°S - 72° S. Zum überwiegenden Teil wurden fahrtbegleitende Messungen durchgeführt. Im Anschluss an die Expedition wurden die Messdaten navigations- und tiefenkorrigiert.

Purpose:

Fahrtbegleitende Messungen zur Unterstützung geologischer und geophysikalischer Untersuchungen, sowie gebietsweise systematische Vermessungen zur Berechnung von digitalen Geländemodellen

Time Period of Content:

Time Period Information:

Range of Dates/Times:

Beginning Date: Februar 2001

Ending Date: April 2001

Currentness Reference:

Aufzeichnung von Fächersonardaten zwischen Februar 2001 und April 2001

Status:

Progress: Messdaten wurden navigation- und tiefeneditiert

Maintenance and Update Frequency: unregelmäßig

Spatial Domain:

Bounding Coordinates:

West Bounding Coordinate: -127.862447

Metadaten von ANT XVIII/5a (Seite 2/5)

East_Bounding_Coordinate: -48.679768
North_Bounding_Coordinate: -51.018537
South_Bounding_Coordinate: -72.504732

Keywords:
Theme:
Theme_Keyword: Bathymetrie
Theme_Keyword: Hydrosweep-Daten
Theme_Keyword: Fächersonarmessungen

Place:
Place_Keyword: Antarktis
Place_Keyword: Südozean

Access_Constraints:
Keine Einschränkungen; Daten sind auf Anfrage unter Angabe des Verwendungszweckes erhältlich.

Use_Constraints:
Wenn die Daten für Publikationen verwendet werden, ist die Quellenangabe erforderlich

Point_of_Contact:
Contact_Information:
Contact_Person_Primary:
Contact_Person: F. Niederjasper
Contact_Organization: Alfred-Wegener-Institut für Polar- und Meeresforschung
Contact_Position: Geosystem- Bathymetrie
Contact_Address:
Address_Type: mailing address
Address: Alfred Wegener Institut, P.O. Box 120161
City: Bremerhaven
Postal_Code: D-27515
Country: Germany
Contact_Voice_Telephone: +49 (0) 471 4831 1227
Contact_Facsimile_Telephone: +49 (0) 471 4831 1149
Contact_Electronic_Mail_Address: fniederjasper@awi-bremerhaven.de
Contact_Instructions: E-Mail Kontakt bevorzugt

Data_Set_Credit: H. W. Schenke, F. Niederjasper, M. Klenke

Native_Data_Set_Environment:
SUN Solaris; Microsoft Windows 2000 Version 5.0 (Build 2195) Service Pack 4; ESRI ArcCatalog 8.3.0.800

Cross_Reference:
Citation_Information:
Originator: Arbeitsgruppe der Bathymetrie
Publication_Date: 2004
Title: Expedition Ant XVIII/5a
Publication_Information:
Publication_Place: Bremerhaven
Publisher: Alfred Wegener Institut für Polar- und Meeresforschung

Data_Quality_Information:

Attribute_Accuracy:

Attribute_Accuracy_Report:

Die Tiefendaten wurden teilweise mit automatischen Filtern und teilweise von Hand bearbeitet.

Im Allgemeinen variiert die Messgenauigkeit innerhalb eines Messpunktes. Dabei liegt das mittlere Drittel des Strahlenbündels eines Footprints innerhalb der Genauigkeit von 0,5% der tatsächlichen Wassertiefe, während die äußeren zwei Drittel innerhalb der Genauigkeit von 1% der tatsächlichen Wassertiefe liegen. Dies gilt für einen Öffnungswinkel von 90°.

Die Positionsgenauigkeit ist abhängig von der Satellitenkonstellation.

ANHANG C. METADATEN

Metadaten von ANT XVIII/5a (Seite 3/5)

Positional Accuracy:

Horizontal_Positional_Accuracy:

Horizontal_Positional_Accuracy_Report:

Positionierung erfolgt auf der Grundlage von Trimble GPS-Empfängern und Inertialsystemen (MINS laser system). Die Positionsgenauigkeit beträgt durchschnittlich 10m.

Vertical_Positional_Accuracy:

Vertical_Positional_Accuracy_Report:

Im Allgemeinen variiert die Messgenauigkeit innerhalb eines Multibeam-Messpunktes. Dabei liegt das mittlere Drittel des Strahlenbündels eines Footprints innerhalb der Genauigkeit von 0,5% der tatsächlichen Wassertiefe, während die äußeren zwei Drittel innerhalb der Genauigkeit von 1% der tatsächlichen Wassertiefe liegen. Dies gilt für einen Öffnungswinkel von 90°.

Lineage:

Source_Information:

Source_Citation:

Citation_Information:

Originator:

Bathymetrie des Alfred Wegener Institutes für Polar und Meeresforschung

Publication_Date: April 2004

Title: Ant XVIII/5a

Type_of_Source_Media: als SURF-Daten verfügbar

Source_Time_Period_of_Content:

Source_Currentness_Reference: Februar 2002 - April 2001

Source_Citation_Abbreviation: Ant XVIII/5a

Process_Step:

Process_Description:

Navigationskorrektur mit HYDROMAP ONLINE; Tiefendatenkorrektur mit Caris HIPS;

Process_Date: 2001

Spatial_Data_Organization_Information:

Direct_Spatial_Reference_Method: Vector

Point_and_Vector_Object_Information:

SDTS_Terms_Description:

SDTS_Point_and_Vector_Object_Type: String

Point_and_Vector_Object_Count: 11

Spatial_Reference_Information:

Horizontal_Coordinate_System_Definition:

Planar:

Map_Projection:

Map_Projection_Name: Stereographic_South_Pole

Planar_Coordinate_Information:

Planar_Coordinate_Encoding_Method: coordinate pair

Coordinate_Representation:

Abscissa_Resolution: 0.004657

Ordinate_Resolution: 0.004657

Planar_Distance_Units: meters

Geodetic_Model:

Horizontal_Datum_Name: D_WGS_1984

Ellipsoid_Name: WGS_1984

Semi-major_Axis: 6378137.000000

Denominator_of_Flattening_Ratio: 298.257224

Vertical_Coordinate_System_Definition:

Metadaten von ANT XVIII/5a (Seite 4/5)

Altitude_System_Definition:
Altitude_Resolution: 1.000000
Altitude_Encoding_Method:
Explicit elevation coordinate included with horizontal coordinates
Depth_System_Definition:
Depth_Datum_Name: Mittlerer Meeresspiegel
Depth_Distance_Units: Decimal Degree

Entity_and_Attribute_Information:
Detailed_Description:
Entity_Type:
Entity_Type_Label: Bathymetry.BATHY.AntXVIII5a
Attribute:
Attribute_Label: OBJECTID
Attribute_Definition: Internal feature number.
Attribute_Definition_Source: ESRI
Attribute_Domain_Values:
Unrepresentable_Domain:
Sequential unique whole numbers that are automatically generated.
Attribute:
Attribute_Label: CRUISEID
Attribute:
Attribute_Label: STARTDATE
Attribute:
Attribute_Label: ENDDATE
Attribute:
Attribute_Label: STARTHARBOR
Attribute:
Attribute_Label: ENDHARBOR
Attribute:
Attribute_Label: INSTITUT
Attribute:
Attribute_Label: SHIPNAME
Attribute:
Attribute_Label: CRUISENAME
Attribute:
Attribute_Label: CHIEFSCIENTIST
Attribute:
Attribute_Label: NAVPLATFORM
Attribute_Definition: <<http://batp8/website/Antarktis>>
Attribute:
Attribute_Label: DEPTHPLATFORM
Attribute_Definition: <<http://batp8/website/Antarktis>>
Attribute:
Attribute_Label: METADATA
Attribute_Definition: <<http://batp8/website/Antarktis>>
Attribute:
Attribute_Label: RAWDATA
Attribute_Definition: <<http://batp8/website/Antarktis>>
Attribute:
Attribute_Label: REPORT
Attribute_Definition: <<http://batp8/website/Antarktis>>
Attribute:
Attribute_Label: SHAPE
Attribute_Definition: Feature geometry.
Attribute_Definition_Source: ESRI
Attribute_Domain_Values:

ANHANG C. METADATEN

Metadaten von ANT XVIII/5a (Seite 5/5)

Unrepresentable_Domain: Coordinates defining the features.

Attribute:

Attribute_Label: SHAPE.len

Overview_Description:

Distribution_Information:

Distributor:

Contact_Information:

Contact_Person_Primary:

Contact_Person: F. Niederjasper

Contact_Organization: Alfred Wegener Institut für Polar- und Meeresforschung

Contact_Position: Geosystem-Bathymetry

Contact_Address:

Address_Type: mailing address

Address: Alfred Wegener Institut, P.O. Box 12 01 61

City: Bremerhaven

Postal_Code: D-27515

Country: Germany

Contact_Voice_Telephone: +49 (0) 471 4831 1227

Contact_Facsimile_Telephone: +49 (0) 471 4831 1149

Contact_Electronic_Mail_Address: fniederjasper@awi-bremerhaven.de

Contact_Instructions: E-Mail Kontakt bevorzugt

Resource_Description: Downloadable Data

Standard_Order_Process:

Digital_Form:

Digital_Transfer_Information:

Format_Name: SURF Format

Fees: Keine

Metadata_Reference_Information:

Metadata_Date: 20040417

Metadata_Contact:

Contact_Information:

Contact_Organization_Primary:

Contact_Organization: Alfred-Wegener-Institut für Polar- und Meeresforschung

Contact_Person: F. Niederjasper

Contact_Position: Geosystem-Bathymetrie

Contact_Address:

Address_Type:

REQUIRED: The mailing and/or physical address for the organization or individual.

City: REQUIRED: The city of the address.

State_or_Province: REQUIRED: The state or province of the address.

Postal_Code: REQUIRED: The ZIP or other postal code of the address.

Contact_Voice_Telephone: +49 (0) 471 4831 1227

Metadata_Standard_Name: FGDC Content Standards for Digital Geospatial Metadata

Metadata_Standard_Version: FGDC-STD-001-1998

Metadata_Time_Convention: local time

Metadata_Extensions:

Online_Linkage: <<http://www.esri.com/metadata/esriprof80.html>>

Profile_Name: ESRI Metadata Profile

Generated by [mp](#) version 2.7.33 on Sat Apr 17 01:06:55 2004

D Javascript

Javascript-Funktion 'showSelect()' (Seite 1/4)

```
// show selected set as an html table
function showSelect(returnString) {
    var anyLayer;
    var layerName = "";
    var msgString = "";
    anyLayer = parent.mapFrame.IMSMap.getSelectedLayer();
    if (anyLayer != null) {
        if (returnString=="") returnString =
            parent.mapFrame.IMSMap.getSelectedLayer().getSelectionSet();
        //Win1.document.write("layerName = anyLayer.getName()");
        layerName = anyLayer.getName();
        var bigString = new String(returnString);
        if (bigString == "") {
            msgString = parent.msgList[10] + layerName;
            alert(msgString);
        }
        else {
            //'theList[FELD]' enthält je Feld eine Datenzeile des
            //Ergebnisdatensatzes;
            //Feld[0] enthält alle Spaltennamen
            var theList = bigString.split("|");
            //Anzahl der Ergebnisdatensätze
            var numRecs = theList.length - 1;
            //'theFields[Feld] enthält je Feld einen Spaltennamen
            //(z.B. BATHYMETRY.BATHY.CRUISE.CRUISEID)
            var theFields = theList[0].split("::");
            // Anzahl der Spaltenparameter
            var numFields = theFields.length;
            var bColor = textBackColor;
            var thisValue = "";
            var charPos = -1;
            var sendValue = "";
            if (dataWindow == 2) {
                // next line for placing returned data in a separate window
                Win1 = open("", "DataWindow", "toolbar,scrollbars,resizable,
                    width=800,height=400");
                dataTarget = " target='targeted'";
            }
            else {
                // next 2 lines for placing returned data in textFrame
                Win1 = parent.textFrame;
                Win1.document.open();
            }
        }
    }
}
```

Javascript-Funktion 'showSelect()' (Seite 2/4)

```
//Beginn der HTML-Seite
Win1.document.write("<body leftmargin=0 topmargin=0 bgcolor='"
+ textBackColor + "' text='" + textForeColor + "' link='"
+ textLinkColor + "' vlink='" + textLinkColor + "'>");
Win1.document.write("<br><div align='center'><font face='Arial'
size='-1'><b>" + layerName + "</b></font><br><br>");
Win1.document.write("<table border='1' width='100%'
cellspacing='0' cellpadding='2'>");

//Tabellenkopf
Win1.document.write("<tr bgcolor='" + bColor + "'><th><FONT
FACE='Arial' SIZE='-2'><b>Rec ? of " + numRecs + "</b></FONT></th>");
for (var j=1;j < theFields.length; j++) {

    //teilt den Verbindungsnamen z.B.: BATHYMETRY.BATHY.CRUISE.CRUISEID
    //in BATHYMETRY,BATHY,CRUISE,CRUISEID
    //nur der letzte Name wird angezeigt
    var theSpalte = theFields[j].split(".");
    var numSpalte = theSpalte.length;//Anzahl der Spaltenparameter z.B.
    //BATHYMETRY.BATHY.CRUISE.CRUISEID == 4;
    Win1.document.write("<th><FONT FACE='Arial' SIZE='1'><b>"
+ theSpalte[numSpalte-1] + "</b></FONT></th>");
}
Win1.document.write("</tr>");//Tabellkopf beendet

var anzahl = 0;
var vergleichswert = 0;
//Tabellenrumpf
//beginnt ab '1', da Feld[0] die Spaltennamen enthält
for (var i=1;i < theList.length; i++) {

    //'theValues[Feld]' enthält je Feld einen Attributwert einer Zeile
    theValues = theList[i].split("::");
    //Vergleich der aktuellen 'CruiseID' mit der des vorhergehenden
    //Datensatzes, wenn ja überspringen
    if (theValues[1] == vergleichswert) {
        continue;
    }
    else {
        vergleichswert = theValues[1];
        //enthält die Anzahl der verschiedenen Datensätze
        anzahl++;
        //wechselt für jeden datensatz die Hintergrundfarbe
        if (bColor == tableBackColor) {
            bColor = textBackColor;
        }
    }
}
```


Javascript-Funktion 'showSelect()' (Seite 3/4)

```

else {
    bColor = tableBackColor;
}
//schreibt die Nummer des Datensatzes
Win1.document.write("<tr bgcolor='" + bColor + "'><td><FONT
    FACE='Arial' SIZE='-3'><b>" + anzahl + "</b></FONT></td>");

//Spaltennamen werden abgefragt, ob sie einen Link enthalten sollen
//beginnt ab '1', da Feld[0] die 'ObjectID' enthält
for (var j=1;j < theFields.length; j++) {

    //teilt den Verbindungsnamen z.B.: BATHYMETRY.BATHY.CRUISE.CRUISEID
    //in BATHYMETRY,BATHY,CRUISE,CRUISEID
    //der letzte Wert ist der eigentliche Spaltenname, dieser wird
    //in der 'case'-Schleife abgefragt
    var theSpalte = theFields[j].split(".");
    //Anzahl der Spaltenparameter
    //z.B. BATHYMETRY.BATHY.CRUISE.CRUISEID == 4;
    var numSpalte = theSpalte.length;

    //konkreter Attributwert einer Zeile
    thisValue = theValues[j];
    sendValue = "<td><font face='Arial' size='-3'>";

    switch (theSpalte[numSpalte-1]) {
        case "STARTDATE": //Damit wird die Angabe der Uhrzeit unterdrückt
            var startdatum = thisValue.split(" ");
            sendValue = sendValue + startdatum[0] + "</a></font></td>";
            break
        case "ENDDATE"://Damit wird die Angabe der Uhrzeit unterdrückt
            var enddate = thisValue.split(" ");
            sendValue = sendValue + enddate[0] + "</a></font></td>";
            break
        //Damit wird die Angabe der Uhrzeit unterdrückt
        case "DATE_OF_COMPLETION":
            var compdate = thisValue.split(" ");
            sendValue = sendValue + compdate[0] + "</a></font></td>";
            break
        case "NAVPLATFORM":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Platform/NavPlatform.pdf'>"
                + thisValue + "</a></font></td>";
            break
    }
}

```

Javascript-Funktion 'showSelect()' (Seite 4/4)

```
        case "DEPTHPLATFORM":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Platform/DepthPlatform.pdf'>"
                + thisValue + "</a></font></td>";
            break
        case "METADATA":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Metadata/" + thisValue + "'>"
                + thisValue + "</a></font></td>";
            break
        case "RAWDATA":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Rawdata/" + thisValue + "/">"
                + thisValue + "</a></font></td>";
            break
        case "REPORT":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Report/" + thisValue + "'>"
                + thisValue + "</a></font></td>";
            break
        case "DATASETS":
            sendValue = sendValue + "<a target='_new | _blank'
                href='http://batp8/website/Data/Dataset/" + thisValue + "/">"
                + thisValue + "</a></font></td>";
            break
        default:
            sendValue = sendValue + thisValue + "</font></td>";
    }
    Win1.document.write(sendValue);
} //Ende for-Schleife: Internet-Links
Win1.document.write("</tr>"); //Abschluss einer Zeile
} //Ende else: eine anderer Datensatz
} //Ende for-Schleife, Tabellenrumpf
//Abschluss des Tabellenrumpfes und der HTML-Seite
Win1.document.write("</table></div></body></html>");
Win1.document.close();
Win1.focus();
Win1 = null;
} //Ende else: bigString<>""
} //Ende if: anyLayer!=null
else {
    msgString = parent.msgList[9];
    alert(msgString);
}
}
```


E Inhalt der beigefügten CD

Inhalt der beigefügten CD

- LaTeX Files und erforderliche Packages
- Abbildungen
- C++Programme
 - NakKonverter
 - AsciiKonverter
- Diplomarbeit.pdf
- Webseite 'Antarktis'
 - Javascript-Funktionen
 - HTML-Dateien
- AntXVIII5a.htm (Metadatenfile nach CSDGM 2)

Eidesstattliche Erklärung

Hiermit erkläre ich, Sebastian Röber, an Eides statt, dass ich die vorliegende Arbeit selbstständig angefertigt habe und keine weiteren außer den angegebenen Quellen und Hilfsmittel verwendet habe.

Bremerhaven, 22.04.2004

